



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**RANCANG BANGUN SISTEM NAVIGASI INDOOR
BERBASIS INTEGRASI SYMBOLIC LOCATION MODEL
DAN WI-FI BASED POSITIONING SYSTEM UNTUK STUDI
KASUS PADA GEDUNG BERTINGKAT**

FARIS MUHAMMAD ASMAWI
NRP 5111 100 157

Dosen Pembimbing
Waskitho Wibisono S.Kom. M.Eng, Ph.D.
Hudan Studiawan S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**RANCANG BANGUN SISTEM NAVIGASI INDOOR
BERBASIS INTEGRASI SYMBOLIC LOCATION
MODEL DAN WIFI BASED POSITIONING SYSTEM
UNTUK STUDI KASUS PADA GEDUNG
BERTINGKAT**

FARIS MUHAMMAD ASMAWI
NRP 5111 100 157

Dosen Pembimbing
Waskitho Wibisono S.Kom. M.Eng, Ph.D.
Hudan Studiawan S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

INDOOR NAVIGATION SYSTEM DESIGN BASED ON INTEGRATION OF SYMBOLIC LOCATION MODEL AND WIFI BASED POSITIONING SYSTEM FOR MULTI LEVEL BUILDING CASE

FARIS MUHAMMAD ASMAWI
NRP 5111 100 157

Dosen Pembimbing
Waskitho Wibisono S.Kom. M.Eng, Ph.D.
Hudan Studiawan S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM NAVIGASI INDOOR BERBASIS INTEGRASI SYMBOLIC LOCATION MODEL DAN WIFI BASED POSITIONING SYSTEM UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

FARIS MUHAMMAD ASMAWI
NRP. 5111 100 157

Disetujui oleh Pembimbing Tugas Akhir

1. Waskitho Wibisono, S.Kom., M.Eng, Ph.D.
NIP: 197410222000031001 (Pembimbing 1)
2. Hudan Studiawan, S.Kom., M.Kom.
NIP: 198705112012121003 (Pembimbing 2)

SURABAYA
JULI, 2017

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM NAVIGASI INDOOR BERBASIS INTEGRASI SYMBOLIC LOCATION MODEL DAN WIFI BASED POSITIONING SYSTEM UNTUK STUDI KASUS PADA GEDUNG BERTINGKAT

Nama Mahasiswa : Faris Muhammad Asmawi
NRP : 5111 100 157
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Waskitho Wibisono S.Kom. M.Eng,
Ph.D.
Dosen Pembimbing 2 : Hudan Studiawan S.Kom., M.Kom.

Abstrak

Kemajuan teknologi informasi saat ini memudahkan setiap orang dalam kegiatannya masing-masing. Saat ini melakukan navigasi tidak perlu menggunakan selembar peta lagi tetapi cukup mengakses GPS ataupun penyedia jasa navigasi lainnya. Teknologi navigasi saat ini sudah sangat maju untuk navigasi di luar ruangan tetapi untuk di dalam ruangan, belum ada standar khusus dan belum terlalu umum digunakan. Tempat umum di Indonesia, khususnya gedung bertingkat, belum menggunakan teknologi sebagai alat navigasinya dan hanya menggunakan selebaran atau peta denah saja. Saat ini masyarakat sudah banyak yang memiliki smartphone sehingga mudah untuk berbagi atau mencari tempat yang diinginkan.

Ketika di dalam gedung bertingkat, kita tidak bisa menggunakan satelit seperti GPS untuk mendapatkan posisi pengunjung. Untuk mendapatkan posisi di dalam ruangan, kekuatan sinyal Wi-Fi akan digunakan sebagai penentu lokasi keberadaan pengunjung. Setiap lokasi yang ada akan memiliki beragam kekuatan sinyal dari Wi-Fi yang berbeda, sehingga setiap lokasi akan unik karena kekuatan sinyal Wi-Fi akan berbeda. Selain menentukan posisi, diperlukan model untuk lokasi-lokasi yang ada sehingga memudahkan pengunjung untuk menentukan

posisinya dan melakukan navigasi ke lokasi lainnya. Untuk membuat sistem navigasi gedung bertingkat diperlukan integrasi keduanya, yaitu penentuan posisi dan pemodelan lokasi. Dalam kasus ini, informasi lokasi destinasi berguna bagi pengunjung yang belum pernah memasuki sebuah gedung dan ingin mendapatkan navigasi yang efisien ke tujuannya. Selain membantu pengguna melakukan navigasi di dalam gedung, pengguna juga bisa berbagi lokasinya kepada pengguna lain jika diinginkan.

Dalam tugas akhir ini, Indoor Positioning berbasis Wi-Fi akan digunakan untuk menentukan posisi menggunakan kekuatan sinyal beberapa Wi-Fi yang berbeda dengan algoritma K Nearest Neighbor dan Euclidean Distance. Symbolic Model digunakan untuk memodelkan sebuah gedung bertingkat dan dapat menangani kasus-kasus yang ada dalam memodelkan lokasi, seperti overlap, mengandung dan terhubung. Dari hasil pengujian, pengguna dapat melakukan pencarian posisi dengan tingkat akurasi sebesar 86.66% dan dapat melakukan navigasi ke setiap lokasi yang ada di Jurusan Teknik Informatika.

Kata kunci: Indoor Positioning, Symbolic Location Model, Wi-fi, Gedung Bertingkat, Navigasi.

INDOOR NAVIGATION SYSTEM DESIGN BASED ON INTEGRATION OF SYMBOLIC LOCATION MODEL AND WIFI BASED POSITIONING SYSTEM FOR MULTI LEVEL BUILDING CASE

Student's Name : Faris Muhammad Asmawi
Student's ID : 5111 100 157
Department : Informatics Engineering, FTIF-ITS
First Advisor : Waskitho Wibisono S.Kom. M.Eng,
Ph.D.
Second Advisor : Hudan Studiawan S.Kom., M.Kom.

Abstract

The advancement of information technology today makes it easier for everyone in their activities. Currently navigating does not need to use a map again but simply access GPS or other navigation service providers. Navigation technology is now very advanced for outdoor navigation but for indoors, there is no special standard and not too commonly used. Public places in Indonesia, especially high rise buildings, do not use technology as a navigation tool and only use leaflets or simple map. Today many people have smartphones, so it is easy to find position or navigate to the desired place.

When inside a multi-story building, we can not use satellites like GPS to get the visitor's position. To get a position in the room, the Wi-Fi signal strength will be used as alternative for determinant of where visitors are located. Each location will have different signal strengths from different Wi-Fi, so each location will be unique because the strength of the Wi-Fi signal will be different. In addition to positioning, modeling is required for existing locations to make it easier for visitors to define their positions and

navigate to other locations. To create a multilevel building navigation system, requires integration of both positioning and location modeling. In this case, destination location information is useful for visitors who have never entered a building and want to get efficient navigation to their destination. In addition to helping users navigate within the building, users can also share their location with other users if desired.

In this final project, Wi-Fi based Indoor Positioning will be used to determine the position of using multiple Wi-Fi signal strengths different from K Nearest Neighbor and Euclidean Distance algorithms. The Symbolic Model is used to model a multi-storey building and can handle cases that exist in modeling locations, such as overlapping, containing and connected. From the test results, users can perform position search with an accuracy rate of 86.66% and can navigate to any location in the Department of Informatics.

Keywords: Indoor Positioning, Symbolic Location Model, Wi-fi, Multi-level building, Navigation.

KATA PENGANTAR

Alhamdulillah, puji syukur kepada Allah SWT atas segala karunia dan rahmat-Nya yang tak terhingga sehingga penulis dapat menyelesaikan Tugas Akhir ini. Shalawat serta salam tak lupa diucapkan kepada Rasulullah SAW, Keluarga, Sahabat dan para pengikutnya hingga akhir zaman.

Tugas akhir ini dapat terselesaikan dikarenakan banyaknya bantuan serta doa yang ditujukan untuk penulis, oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya pada kesempatan ini kepada:

1. Orang tua penulis, Asngadi dan Noneu Andryani yang selalu memberikan dukungan, semangat, kasih sayang dan doa.
2. Bapak Waskitho Wibisono selaku dosen pembimbing 1 dan Bapak Hudan Studiawan. selaku dosen pembimbing 2, yang telah memberikan kepercayaan, dukungan, bimbingan dan nasihat untuk penulis.
3. Segenap staf dan dosen pengajar di Jurusan Teknik Informatika ITS yang membantu penulis semasa perkuliahan.
4. Teman-teman kontrakan yang menemani penulis: Toto, Rahman, Tommy, Ruslan, Punggi, Bestama, Andrie, Yunus, Tev dan Ipir.
5. Teman-teman lhoktuan yang memberikan bantuan dan semangat kepada penulis: Adi, Rifi, Fahmi, Satrio dan Bachmid.
6. Teman-teman noragret yang membuat keseharian penulis lebih hidup dan saling membimbing satu sama lain: Rahmat, Tegar dan Adi.
7. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tugas Akhir ini.

Semoga Allah SWT memberkati dan membalas semua kebaikan yang telah dilakukan. Penulis menyadari masih banyak yang dapat dikembangkan pada Tugas Akhir ini. Oleh karena itu, penulis menerima setiap masukan dan kritik yang diberikan. Semoga Tugas Akhir ini dapat memberikan manfaat.

Surabaya, Mei 2017

Faris Muhammad Asmawi

DAFTAR ISI

LEMBAR PENGESAHAN	vii
Abstrak	ix
Abstract	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xx
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat	3
1.5 Metodologi	3
1.6 Sistematika Penyusunan Laporan.....	4
2 BAB II TINJAUAN PUSTAKA.....	7
2.1 Indoor Positioning System	7
2.2 Wireless Fidelity (Wi-Fi)	7
2.3 Algoritma K Nearest Neighbor	7
2.3.1 Euclidean Distance	8
2.4 Symbolic Location Model	9
2.5 Algoritma Dijkstra	10
2.6 Java dan Android.....	10
2.7 WifiManager Android	10
2.8 Preprocessor Hypertext (PHP)	11
2.9 MySQL.....	11
3 BAB III PERANCANGAN PERANGKAT LUNAK.....	13
3.1 Deskripsi Umum Sistem.....	13
3.2 Arsitektur Umum Sistem.....	13
3.3 Perancangan Data.....	15
3.3.1 Data Pelatihan dan Pengeluaran.....	15
3.3.2 Pemodelan Lokasi	16

3.4	Perancangan Penentuan Posisi Pengguna Menggunakan Wi-Fi	17
3.4.1	Desain Secara Umum	17
3.4.2	Tahap Pra-Proses Data	17
3.4.3	Tahap Pencarian Nilai K yang Optimal.....	18
3.4.4	Tahap Perhitungan untuk Menentukan Posisi Pengguna. .	18
3.5	Perancangan Algoritma untuk Melakukan Navigasi	19
3.6	Integrasi Model Lokasi Simbolik dan Indoor Positioning System	19
3.7	Use Case Sistem	20
3.8	Perancangan Antarmuka Sistem.....	21
3.8.1	Perancangan Antarmuka Melihat Lokasi Saat Ini	22
3.8.2	Perancangan Antarmuka Melihat Lokasi yang Tersedia...	23
3.8.3	Perancangan Antarmuka Navigasi	23
4	BAB IV IMPLEMENTASI PERANGKAT LUNAK.....	25
4.1	Lingkungan Implementasi.....	25
4.1.1	Lingkungan Implementasi Perangkat Keras.....	25
4.1.2	Lingkungan Implementasi Perangkat Lunak.....	25
4.2	Implementasi Basis Data	26
4.2.1	Implementasi Struktur Basis Data	26
4.2.2	Implementasi Query	28
4.3	Implementasi Pengambilan Data Latih.....	31
4.4	Inspeksi Sampel Data dan Pemilihan Nilai K optimal	31
4.5	Implementasi Menentukan Posisi.....	33
4.6	Implementasi Navigasi	36
4.7	Implementasi Menggambar Posisi dan Navigasi pada Peta ..	36
4.7.1	Implementasi Menggambar Posisi	36
4.7.2	Implementasi Menggambar Jalan Navigasi.....	37
4.8	Implementasi Antarmuka	38
4.8.1	Antarmuka Lokasi Saat Ini	38
4.8.2	Antarmuka Daftar Lokasi yang Tersedia	39
4.8.3	Antarmuka Navigasi.....	40
5	BAB V UJI COBA DAN EVALUASI.....	43
5.1	Lingkungan Uji Coba	43

5.2	Data Uji Coba.....	44
5.3	Skenario & Hasil Uji Coba.....	44
5.3.1	Pengujian Fungsionalitas.....	44
5.3.2	Uji Coba Akurasi Deteksi Posisi.....	54
5.3.3	Uji Coba Performa dari Aplikasi.....	56
5.4	Evaluasi	57
6	BAB VI KESIMPULAN DAN SARAN.....	59
6.1	Kesimpulan	59
6.2	Saran	59
	DAFTAR PUSTAKA.....	61
7	LAMPIRAN A DAFTAR LOKASI YANG DIGUNAKAN.....	63
A.	Peta Lokasi yang Digunakan.....	63
B.	Daftar Lokasi yang Digunakan	68
8	LAMPIRAN B PERHITUNGAN DALAM MENENTUKAN K YANG OPTIMAL.....	70
	BIODATA PENULIS	75

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Model Lokasi Simbolik	9
Gambar 2.2 Level dari Detil Jenis Kombinasi	9
Gambar 3.1 Perancangan Aristektur Umum Sistem.....	14
Gambar 3.2 Model Lokasi dari Jalan Utara Lantai 2.....	16
Gambar 3.3 Posisi dalam melakukan scanning untuk setiap segmen	18
Gambar 3.4 Diagram Use Case	21
Gambar 3.5 Perancangan Antarmuka Melihat Lokasi Saat Ini ...	22
Gambar 3.6 Perancangan Navigasi.....	23
Gambar 3.7 Perancangan Antarmuka Ketika Melakukan Navigasi dari salah satu kasus	24
Gambar 4.1 Tampilan Antarmuka Pengambilan Data Latih	31
Gambar 4.2 Tampilan Antarmuka Lokasi Saat Ini.....	39
Gambar 4.3 Tampilan Antarmuka Lokasi yang Tersedia.....	40
Gambar 4.4 Tampilan Antarmuka Navigasi.....	41
Gambar 5.1 Hasil Uji Coba Menampilkan Lokasi yang Tersedia	45
Gambar 5.2 Tampilan Uji Coba Melihat Gambar Peta dari Lokasi yang Diinginkan	46
Gambar 5.3 Hasil Uji Coba Menampilkan Lokasi Saat Ini	47
Gambar 5.4 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107	49
Gambar 5.5 Tampilan Detil Peta Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107	50
Gambar 5.6 Tampilan Jalan Tangga 3 ke Jalan Timur Lantai 1 Melalui Plasa Baru 1 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107	51
Gambar 5.7 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru 3 Menuju IF107.....	52
Gambar 5.8 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru 3 Menuju IF107.....	53

DAFTAR TABEL

Tabel 3.1 Tabel Deskripsi <i>Use Case</i>	21
Tabel 4.1 Tabel Perhitungan K dengan voting hasil.....	33
Tabel 5.1 Prosedur Uji Coba Lokasi yang Tersedia	44
Tabel 5.2 Prosedur Uji Coba Menampilkan Lokasi Saat Ini	48
Tabel 5.3 Prosedur Uji Coba Menampilkan Detail Aduan	53
Tabel 5.4 Uji Coba Akurasi Deteksi Posisi	55
Tabel 5.5 Uji Coba Performa Mencari Posisi saat Ini	56
Tabel 5.6 Uji Coba Performa Melakukan Navigasi.....	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi saat ini memudahkan setiap orang dalam kegiatannya masing-masing. Saat ini melakukan navigasi tidak perlu menggunakan selembar peta lagi tetapi cukup mengakses GPS ataupun penyedia jasa navigasi lainnya. Teknologi navigasi saat ini sudah sangat maju untuk navigasi di luar ruangan tetapi untuk di dalam ruangan, belum ada standar khusus dan belum terlalu umum digunakan [1]. Tempat umum di Indonesia, khususnya gedung bertingkat, belum menggunakan teknologi sebagai alat navigasinya dan hanya menggunakan selebaran atau peta denah saja. Saat ini masyarakat sudah banyak yang memiliki smartphone sehingga mudah untuk berbagi atau mencari tempat yang diinginkan.

Ketika di dalam gedung bertingkat, kita tidak bisa menggunakan satelit seperti GPS untuk mendapatkan posisi pengunjung. Untuk mendapatkan posisi di dalam ruangan, kekuatan sinyal Wi-Fi akan digunakan sebagai penentu lokasi keberadaan pengunjung. Setiap lokasi yang ada akan memiliki beragam kekuatan sinyal dari Wi-Fi yang berbeda, sehingga setiap lokasi akan unik karena kekuatan sinyal Wi-Fi akan berbeda. Selain menentukan posisi, diperlukan model untuk lokasi-lokasi yang ada sehingga memudahkan pengunjung untuk menentukan posisinya dan melakukan navigasi ke lokasi lainnya. Untuk membuat sistem navigasi gedung bertingkat diperlukan integrasi keduanya, yaitu penentuan posisi dan pemodelan lokasi. Dalam kasus ini, informasi lokasi destinasi berguna bagi pengunjung yang belum pernah memasuki sebuah gedung dan ingin mendapatkan navigasi yang efisien ke tujuannya. Selain membantu pengguna melakukan navigasi di dalam gedung, pengguna juga bisa berbagi lokasinya kepada pengguna lain jika diinginkan.

Dalam tugas akhir ini, Indoor Positioning berbasis Wi-Fi akan digunakan untuk menentukan posisi menggunakan kekuatan

sinyal beberapa Wi-Fi yang berbeda. Model lokasi yang digunakan adalah *Symbolic Location Model*, selain mudah dibaca, model tersebut memiliki kemudahan untuk menentukan posisi terdekat karena model pengelompokannya yang general [2]. *Symbolic Model* dapat memodelkan sebuah gedung bertingkat dan dapat menangani kasus-kasus yang ada dalam memodelkan lokasi, seperti overlap, mengandung dan terhubung.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat *location model* untuk kasus gedung bertingkat menggunakan *symbolic location*?
2. Bagaimana cara mengintegrasikan *indoor positioning system* dengan *symbolic location*?
3. Bagaimana cara melakukan navigasi untuk studi kasus di dalam gedung bertingkat?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi ini terdiri dari server dan client, berbasis php dan java untuk client serta server dan MySQL untuk database.
2. Uji coba akan dilakukan di lingkungan Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember sebagai kasus gedung bertingkat.
3. Modul client dijalankan pada sistem operasi Android dengan versi 7 (Nougat).
4. Algoritma yang dipakai dalam mendapatkan posisi di dalam ruangan adalah algoritma *K Nearest Neighbor* dan model lokasi yang digunakan adalah *Symbolic Location Model*.

1.4 Tujuan dan Manfaat

Tujuan dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi yang dapat menentukan posisi ketika berada di dalam gedung.
2. Membuat model untuk gedung bertingkat dengan Symbolic Location.
3. Melakukan navigasi ketika berada di dalam ruangan.

Manfaat dari hasil pembuatan tugas akhir ini adalah mempermudah dan meningkatkan pengalaman pengguna dalam melakukan navigasi di dalam gedung bertingkat.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan Tugas Akhir.

a. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada Tugas Akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik Tugas Akhir ini, di antaranya mengenai:

- a. *Indoor Positioning*
- b. Algoritma *K Nearest Neighbor*.
- c. Algoritma *Dijkstra*.
- d. Model *Symbolic Location*.
- e. *Scanning* kekuatan sinyal Wi-Fi pada perangkat Android.
- f. Analisa pengiriman dan pengolahan data dari perangkat Android ke server.

b. Desain Sistem

Tahap ini merupakan perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan menggunakan teori, metode, dan informasi

yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses perancangan sistem.

c. Implementasi

Implementasi merupakan tahap penerapan bagaimana sinyal Wi-fi yang telah di *scanning* diproses oleh server. Pemrosesan yang dilakukan adalah menentukan posisi pengguna dan memberikan panduan navigasi untuk pengguna jika ingin menuju ke lokasi lain.

d. Uji Coba dan Evaluasi

Pengujian aplikasi ini dapat dilakukan antara lain sebagai berikut:

1. Menentukan lokasi pengguna berdasarkan sinyal Wi-fi yang telah di *scanning*.
2. Menggunakan sistem navigasi untuk pergi ke lokasi yang diinginkan.
3. Aplikasi dapat menunjukkan lokasi-lokasi yang ada dalam model yang telah dibuat.

e. Penyusunan Laporan Tugas Akhir

Tahap ini digunakan untuk membuat laporan Tugas Akhir yang berisi metode, dasar teori, dan hasil yang didapatkan selama pengerjaan Tugas Akhir.

1.6 Sistematika Penyusunan Laporan

Buku Tugas Akhir ini disusun dengan sistematika laporan sebagai berikut:

1. Bab I. Pendahuluan
Bab ini meliputi latar belakang masalah, rumusan permasalahan, batasan masalah, tujuan, dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan laporan Tugas Akhir.
2. Bab II. Tinjauan Pustaka

Bab ini meliputi dasar teori dan penunjang yang berkaitan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

3. Bab III. Perancangan Perangkat Lunak

Bab ini membahas desain dari sistem yang akan dibuat meliputi arsitektur sistem, *use case* sistem, dan perancangan antarmuka sistem.

4. Bab IV. Implementasi

Bab ini membahas implementasi dari desain sistem yang dilakukan pada tahap desain, meliputi *pseudocode* dan implementasi antarmuka dari perangkat lunak.

5. Bab V. Uji Coba dan Evaluasi

Bab ini membahas uji coba dari perangkat lunak yang dibuat dengan melihat keluaran yang dihasilkan oleh perangkat lunak, analisis, dan evaluasi untuk mengetahui kemampuan perangkat lunak.

6. Bab VI. Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan serta saran untuk pengembangan lebih lanjut perangkat lunak.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 *Indoor Positioning System*

Indoor positioning system (IPS) adalah sistem pencarian posisi untuk di dalam ruangan yang berbasis *context-aware* [3]. *Context-aware* adalah sadar dengan situasi sekitar dan secara otomatis beradaptasi dengan konteks tersebut [4]. Sistem ini menggunakan kekuatan sinyal *Wi-Fi* yang digunakan sebagai konteks untuk menentukan posisi dari pengguna. Penggunaan sistem pencarian posisi populer seperti *Global Positioning System* (GPS) untuk mencari posisi di dalam ruangan juga kurang efektif karena sinyal satelit tertutup oleh bangunan.

Dalam tugas akhir ini, dibangun sebuah aplikasi yang menggunakan sinyal *Wi-Fi* untuk mengetahui posisi pengguna ketika berada di dalam ruangan.

2.2 *Wireless Fidelity (Wi-Fi)*

Wireless Fidelity mengacu pada IEEE 802.11 *wireless local area network* yang merupakan teknologi jaringan nirkabel yang digunakan untuk menghubungkan perangkat yang mendukung, seperti laptop, ponsel dan lain-lain. Teknologi ini menjadi populer seiring perkembangan konektivitas IP di ranah publik dan komersial. Beberapa standar 802.11 yang digunakan di publik adalah 802.11a, 802.11b, 802.11g dan 802.11n [5].

2.3 *Algoritma K Nearest Neighbor*

K Nearest Neighbors adalah algoritma yang menyimpan seluruh data yang tersedia dan mengklasifikasikan kasus baru

berdasarkan kesamaan dalam perhitungan (seperti fungsi jarak). K Nearest Neighbors atau KNN telah banyak digunakan untuk estimasi statistic dan mengenali pola.

Algoritma ini mengklasifikasikan sebuah kasus dengan melakukan voting mayoritas dari neighbors nya, dimana kasus diberikan pada kelas yang paling mirip diantara tetangganya yang dihitung berdasarkan fungsi jarak. Jika $K = 1$, maka kasusnya cukup memilih kelas dengan tetangga terdekat [6].

Cara memilih nilai K optimal yang terbaik adalah dengan menginspeksi dahulu data yang ada. Secara umum, nilai K yang besar akan menghasilkan data yang akurat karena mengurangi keseluruhan noise tetapi tetap tanpa jaminan. Validasi silang adalah cara lain secara retrospektif untuk menentukan nilai K yang baik dengan menggunakan dataset yang independen untuk memvalidasi nilai dari K. Dari perhitungan-perhitungan yang pernah dilakukan, nilai optimal dari K berada pada 3 hingga 10, dimana hasil dari klasifikasi akan lebih baik dari 1 *Nearest Neighbor*.

2.3.1 Euclidean Distance

Perhitungan jarak yang akan digunakan adalah perhitungan jarak Euclidean dimana hasil akhir perhitungan akan digunakan untuk *voting* mencari *neighbor* terdekatnya [6].

Berikut adalah persamaan Euclidean:

$$EuclideanDist(x_i, y_i) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

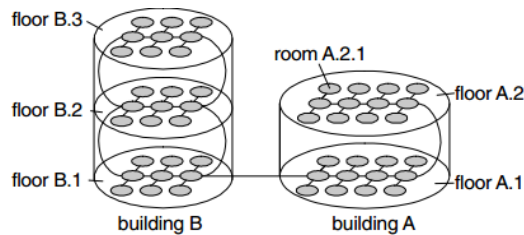
Persamaan 2.1 Euclidean Distance

Pada tugas akhir ini, persamaan akan digunakan untuk mencari posisi pengguna. x_i adalah data kekuatan sinyal Wi-Fi yang telah diinspeksi dan digunakan sebagai acuan, y_i adalah data kekuatan sinyal Wi-Fi saat melakukan pencarian posisi. Hasil dari perhitungan jarak Euclidean akan dipilih yang terendah dan jumlah

perhitungan yang akan di *voting* berdasarkan jumlah K dari K *Nearest Neighbor* yang akan digunakan.

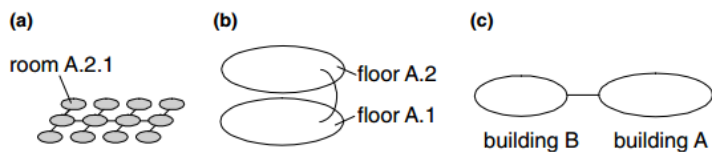
2.4 Symbolic Location Model

Symbolic location model adalah model yang terdiri dari hierarki dari lokasi simbolik, dimodelkan berdasarkan hubungan spasial “mengandung”.



Gambar 2.1 Model Lokasi Simbolik

Model lokasi simbolik seperti gambar 2.2 memiliki kemampuan untuk menunjukkan sebuah detail dari model dan melakukan pembagian ke model yang lebih kecil seperti gambar di bawah ini:



Gambar 2.2 Level dari Detil Jenis Kombinasi

Dalam penerapannya, kita dapat menggunakan untuk mendaftarkan lokasi yang ada sesuai dengan pengelompokannya serta mendaftarkan jarak antara lokasi yang terhubung. Pemodelan lokasi ini mirip dengan graph sehingga algoritma pencarian jarak terdekat dapat dilakukan [2].

2.5 Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma untuk mencari jarak terpendek dari *graph* berarah dengan bobot tidak negatif di sisinya. Cara kerja Algoritma dijkstra memakai strategi *greedy*, dimana pada setiap langkah di pilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul yang sudah terpilih dengan simpul lain yang belum terpilih.

Algoritma Dijkstra membutuhkan parameter tempat asal ,tempat tujuan dan jarak dari jalan yang terhubung. Hasil akhir dari algoritma ini adalah jarak terpendek dari tempat asal ke tempat ujuan beserta rutenya.

Pada tugas akhir ini, algoritma Dijkstra digunakan untuk melakukan navigasi dimana pemodelan lokasi adalah model simbolik yang mirip dengan graph. Pemodelan simbolik memiliki informasi berupa lokasi, hubungan antar lokasi dan jarak hubungan antar lokasi sehingga perhitungan jarak terpendek dapat dilakukan.

2.6 Java dan Android

Java adalah salah satu bahasa pemrograman berbasis objek yang dikembangkan oleh SUN *Microsytem* yang dirilis pada tahun 1995. Java merupakan bahasa pemrograman berbasis objek murni dan dapat dijalankan di berbagai *platform* atau sistem operasi komputer. Hal ini dikarenakan Java tidak seperti bahasa-bahasa pemrograman lainnya, kode sumber (*source code*) dari Java dibaca oleh *runtime* Mesin Virtual Java (*Java Virtual Machine*) sehingga memungkinkan untuk dijalankan di berbagai *platform*.

Android merupakan sistem operasi *mobile* berbasis Linux yang dikembangkan oleh Android, Inc., yang dibeli oleh Google pada tahun 2005. Pengembangan aplikasi Android menggunakan bahasa pemrograman Java [7].

2.7 WifiManager Android

Android memungkinkan aplikasi mengakses koneksi Wi-Fi pada tingkat yang sangat rendah. Aplikasi bisa mengakses hampir semua informasi koneksi wifi.

Informasi yang dapat diakses oleh sebuah aplikasi mencakup kecepatan tautan jaringan yang terhubung, alamat IP, status negosiasi, informasi jaringan lainnya. Aplikasi juga dapat memindai, menambah, menyimpan, mengakhiri dan memulai koneksi Wi-Fi.

Android menyediakan WifiManager API untuk mengelola semua aspek konektivitas Wi-Fi [8]. Pada tugas akhir ini, WifiManager akan mengakses informasi berupa nama *access point*, *mac address* access point dan kekuatan sinyal dari *access point*.

2.8 *Preprocessor Hypertext (PHP)*

PHP merupakan sebuah bahasa pemrograman scripting open source yang dijalankan di bagian server yang dibuat untuk pengembangan aplikasi web [9].

2.9 MySQL

MySQL merupakan sistem manajemen basis data yang bersifat *open source*. MySQL dikembangkan, didistribusikan, dan didukung oleh perusahaan Oracle. MySQL merupakan sistem manajemen basis data yang berfungsi untuk menambah, menyimpan, dan mengakses data yang tersimpan di dalam basis data yang ada di komputer. Basis data MySQL bersifat relasional.

Keuntungan dari MySQL adalah mendukung untuk standar SQL, dapat berjalan di berbagai sistem operasi seperti Windows, Linux, FreeBSD, dan Mac OS. Antar muka MySQL juga tersedia untuk berbagai bahasa pemrograman antara lain C/C++, C#, Java, PHP, Python, dan Ruby [10].

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN PERANGKAT LUNAK

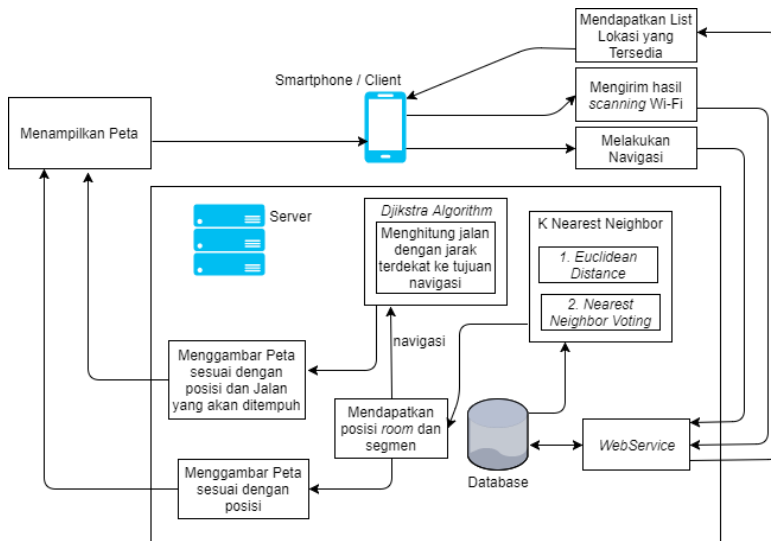
Perancangan merupakan bagian penting dari pembuatan suatu perangkat lunak yang berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses, alur, dan implementasinya.

3.1 Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibangun modul aplikasi *Wireless Indoor Navigation System* yang memiliki kemampuan untuk menentukan posisi pengguna ketika berada di dalam bangunan Jurusan Teknik Informatika ITS dan melakukan navigasi di dalamnya. Hasil akhir dari aplikasi ini adalah menampilkan posisi dari pengguna dan menunjukkan jalan menuju lokasi yang akan dituju pengguna. Aplikasi ini menampilkan lokasi pengguna dengan melakukan *scanning* posisi pengguna dan menunjukkan lokasi yang ingin dituju ketika melakukan navigasi.

3.2 Arsitektur Umum Sistem

Sistem ini dirancang menggunakan *server* sebagai penyimpan dan pengirim data, *client* yang merupakan perangkat bergerak sebagai pengambil data. *Server* merupakan *web service* dan *database server*, sedangkan *client* merupakan modul aplikasi pada *smartphone*. *Client* bertugas untuk mengambil data dan mengirim ke *server*. *Server* yang kemudian memproses dan mengirim hasil proses kembali ke *client*. *Server* bertugas untuk menerima data dan mengirimkan data yang telah diproses kembali ke *client*.



Gambar 3.1 Perancangan Aristektur Umum Sistem

Berdasarkan perancangan arsitektur umum sistem pada Gambar 3.1, *Client smartphone* dapat melakukan pencarian posisi dan melakukan navigasi. *Smartphone* sebagai perangkat bergerak akan mengirimkan data berupa sinyal *Wi-Fi* beserta *mac address* dari *access point* yang telah di *scanning* pada lokasi tertentu.

Data kekuatan sinyal *Wi-Fi* dan *mac address* kemudian di proses oleh server. Sebelum proses, *server* meminta kepada *database* data latih dari setiap posisi yang ada. Setelah data hasil *scanning* dan data latih tersedia, maka dilakukan proses perhitungan dengan pendekatan *K Nearest Neighbor*.

Pada perhitungan untuk mencari posisi, yaitu perhitungan dengan pendekatan *K Nearest Neighbor*, terdapat 2 tahap yaitu menghitung jarak Euclidean antara data hasil *scan* dengan data latih kemudian dilakukan *voting* dari hasil perhitungan jaraknya. *Voting* dilakukan pada hasil perhitungan Euclidean sebanyak nilai K yang optimal. Hasil akhir dari perhitungan adalah estimasi posisi dari pengguna.

Penggambaran pada peta akan dilakukan setelah mendapatkan posisi dari hasil perhitungan. Peta kemudian diberi

penanda dan akan dikirimkan ke *client* untuk ditampilkan. Perbedaan antara pencarian posisi dengan navigasi adalah, pencarian posisi hanya menggambarkan posisi dari user sedangkan navigasi akan melakukan perhitungan untuk menentukan jarak terpendek sebelum digambar. Pada proses navigasi, jalan menuju lokasi yang diinginkan dan posisi pengguna akan digambarkan pada peta.

Perhitungan untuk menentukan jarak terpendek dari posisi pengguna menuju lokasi tertentu dilakukan dengan pendekatan algoritma Dijkstra. Perhitungan dilakukan menggunakan data dari pemodelan lokasi simbolik yang telah dilakukan sebelumnya. Model lokasi simbolik mirip dengan graph, yaitu memiliki data lokasi, lokasi yang terhubung dan jarak antar lokasi yang terhubung.

3.3 Perancangan Data

Perancangan data adalah hal yang penting dalam sistem karena diperlukan data yang tepat agar sistem dapat beroperasi dengan benar. Sistem yang dibuat membutuhkan data masukan dan akan memberikan data keluaran. Data masukan berupa himpunan data yang sebagian akan digunakan sebagai data latih. Data latih ini akan digunakan dalam penentuan segmen dan lokasi ruangan. Data akhir adalah data klasifikasi beserta data aduan yang terdapat pada tiap klasifikasi serta data aduan yang mirip setelah dilakukan deteksi kemiripan aduan pada masing-masing *cluster*.

3.3.1 Data Pelatihan dan Pengeluaran

- **Data masukan**

Data yang digunakan adalah kekuatan sinyal dan *mac address* dari *access point* setiap lokasi yang ada di Teknik Informatika Institut Teknologi Sepuluh Nopember. Data masukan yang telah diinspeksi akan dijadikan data pelatihan yang digunakan sebagai acuan untuk menentukan posisi pengguna.

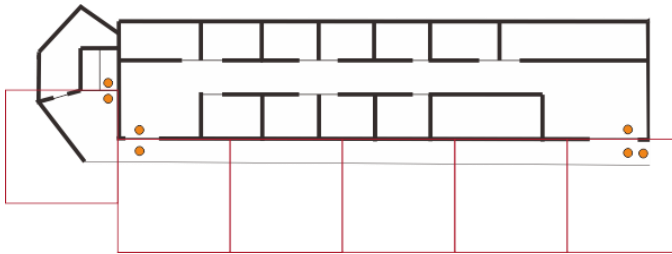
- **Data keluaran**

Data keluaran berupa posisi pengguna yang akan ditampilkan dalam bentuk gambar peta.

3.3.2 Pemodelan Lokasi

3.3.2.1 Pemodelan Lokasi untuk Indoor Positioning System

Lokasi memiliki 2 bagian, yaitu *room* dan segmen. *Room* adalah lokasi yang memiliki cakupan lebih besar seperti lantai atau bagian dari sebuah gedung. Segmen adalah lokasi lebih spesifik dari *Room* yang digunakan untuk mencari posisi yang lebih akurat. Setiap segmen memiliki luas sekitar 3.9 meter x 3.9 meter atau 13 ubin berukuran 30 cm.



Gambar 3.2 Model Lokasi dari Jalan Utara Lantai 2

Pada gambar 3.2, *Room* adalah Jalan Utara Lantai 2 dan persegi dengan garis berwarna merah adalah segmen. Penggunaan model lokasi *symbolic* akan diterapkan antar *Room* dan diberi penghubung serta jarak dengan *Room* lain yang terhubung. Lokasi yang dapat digunakan untuk melakukan navigasi adalah *Room* dengan minimal terdapat 3 *access point* atau lebih.

3.3.2.2 Pemodelan Lokasi Simbolik untuk Navigasi dan Pemetaan

Dalam pemodelan lokasi simbolik, model harus bisa menyimpan lokasi dengan hubungan mengandung. Pada tugas akhir ini, pemodelan yang akan dilakukan adalah menyimpan daftar lokasi yang tersedia, menyimpan lokasi-lokasi yang terhubung pada sebuah lokasi beserta jaraknya dan lokasi yang ada pada peta.

```
'JU1' => ['MUSHOLLA', 'KANTIN', 'IF101',
           'IF102', 'IF103', 'IF104']
```

Kode 3.1 Contoh dari penyimpanan daftar lokasi yang terhubung

Pada penyimpanan jarak antara lokasi, dapat menggunakan model lokasi untuk menyimpan hubungan antar lokasi untuk dijadikan indeks penunjuk jaraknya. Selain itu perlu pengukuran jarak yang baik untuk menghasilkan navigasi ketika menggunakan model lokasi simbolik sebagai basisnya.

<pre>'JU1' => ['100', '200', '150', '100', '150', '200']</pre>

Kode 3.2 Contoh dari penyimpanan jarak antar lokasi berdasarkan daftar lokasi yang terhubung

Untuk menyimpan lokasi yang tersedia pada gambar peta, dapat menggunakan format seperti penyimpanan daftar lokasi yang terhubung. Inti dari penyimpanan lokasi simbolik adalah menampilkan lokasi dengan simbol yang dapat dimengerti manusia dan menunjukkan lokasi yang unik identitasnya.

3.4 Perancangan Penentuan Posisi Pengguna Menggunakan Wi-Fi

Dalam penentuan posisi pengguna, Algoritma *K Nearest Neighbor* digunakan untuk melakukan klasifikasi perhitungan dari *Euclidean Distance* dari sinyal yang telah di *scanning* oleh pengguna.

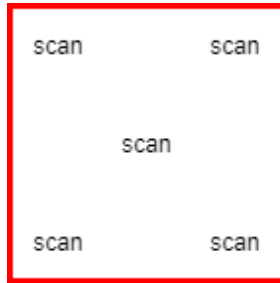
3.4.1 Desain Secara Umum

Perhitungan dilakukan dengan melakukan perbandingan antara kecocokan *access point* dengan data sampel. Kecocokan dilihat dari *mac address* milik *access point* tersebut. Setelah mendapatkan *access point* yang sesuai dengan hasil *scanning*, perhitungan jarak Euclidean akan dilakukan. Setelah melakukan perhitungan jarak, *K Nearest Neighbor* akan dilakukan yaitu dengan melakukan voting dari hasil perhitungan jarak Euclidean sebanyak nilai *K* optimal yang telah dicari sebelumnya.

3.4.2 Tahap Pra-Proses Data

Tahap ini adalah tahap menyiapkan data latih dengan melakukan *scanning* data dari setiap segmen dan *room* yang ada

pada lokasi. Tahap ini juga dilakukan inspeksi pada setiap data latih yang telah digunakan sehingga klasifikasi dalam mencari posisi lebih akurat. Gambar 3.5 adalah posisi pengambilan data latih dari setiap segmen.



Gambar 3.3 Posisi dalam melakukan *scanning* untuk setiap segmen

Inspeksi dilakukan untuk melihat apakah nilai dari hasil scan dapat dijadikan acuan untuk perhitungan dalam mencari ruangan dan segmen. Jika data dari hasil *scanning* terdapat *noise* atau terjadi kesalahan maka data tersebut dapat dihilangkan atau diganti dengan data baru.

3.4.3 Tahap Pencarian Nilai K yang Optimal

Tahap ini merupakan tahap dimana menentukan nilai K yang optimal akan menghasilkan posisi yang lebih akurat. Nilai K optimal dilakukan dengan melakukan perhitungan Euclidean menggunakan data latih dan percobaan dari nilai $K=1$ hingga $K=10$. Setelah perhitungan dilakukan maka dilakukan *voting* hingga menemukan nilai K yang optimal.

3.4.4 Tahap Perhitungan untuk Menentukan Posisi Pengguna.

Tahap ini adalah tahap memproses atau melakukan klasifikasi untuk menentukan posisi dari pengguna berdasarkan data *scanning* yang dikirim oleh pengguna. Setelah menentukan K yang optimal, hasil *scanning* dengan data latih yang dikumpulkan sebelumnya akan digunakan untuk perhitungan Euclidean dan diklasifikasikan menggunakan *K Nearest Neighbor* yang menghasilkan posisi pengguna.

Proses yang terjadi dalam penentuan posisi pengguna adalah setelah melakukan perhitungan jarak Euclidean, akan dilakukan pengurutan dan pemotongan dari hasil perhitungan. Setelah diurutkan. Akan dilakukan *voting* dari *neighbor* sejumlah K. Jika dalam proses *voting*, terdapat ruangan yang sama, maka akan dihitung jumlah kemunculan ruangan dan perhitungan jarak Euclideannya akan dihitung rata-ratanya.

Pemilihan lokasi dalam proses voting dilakukan dengan prioritas jumlah kemunculan ruangan terbanyak kemudian rata-rata jarak terkecil apabila terjadi kesamaan dalam kemunculan ruangan terbanyak.

3.5 Perancangan Algoritma untuk Melakukan Navigasi

Algoritma yang digunakan adalah algoritma Dijkstra dimana algoritma tersebut digunakan untuk pencarian jarak terpendek. Algoritma ini akan menghitung jarak dari lokasi asal hingga lokasi tujuan hingga menemukan jalan dengan lokasi yang terpendek. Algoritma tersebut dapat dilihat pada Kode 3.1.

Input: Node Awal dan Node Tujuan
1 Inisialisasi posisi Node dan jarak tiap edge
2 Selama $i < \text{semua node yang terhubung}$
3 Tambahkan jarak setiap melewati edge pada node yang terhubung.
4 Jika sampai hingga node tujuan simpan jarak dan lokasi yang dilalui.
5 Jika pencarian node melewati batas jarak, berhenti.
6 Jika pencarian jalan menghasilkan jarak yang lebih rendah menuju ke node tujuan, simpan dan jadikan jarak yang lebih kecil sebagai batas.
Output: node yang dilewati dan total jarak yang ditempuh.

Kode 3.3 *Pseudocode* dari Algoritma Dijkstra

3.6 Integrasi Model Lokasi Simbolik dan Indoor Positioning System

Pada perancangan untuk tugas akhir ini, Integrasi dilakukan agar aplikasi dapat melakukan seluruh fungsi yang akan

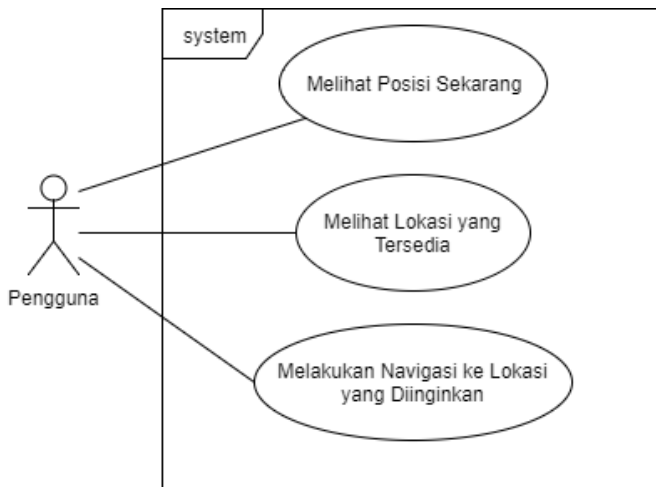
digunakan. Fungsi pencarian posisi tidak akan dapat menampilkan lokasinya jika model lokasi tidak ada dan fungsi navigasi tidak dapat dilakukan jika tidak ada perhitungan untuk menentukan posisi awal pengguna.

Integrasi dilakukan dengan menyimpan seluruh data latihan setiap segmen pada ruangan dari lokasi yang telah di modelkan oleh pemodelan lokasi simbolik. Pada penentuan posisi di dalam ruangan, hasil klasifikasi dari algoritma *K Nearest Neighbor* adalah ruangan yang telah dimodelkan oleh pemodelan simbolik dengan data latihan yang telah diintegrasikan.

Untuk navigasi, integrasi digunakan untuk mencari posisi awal pengguna ketika menggunakan aplikasi. Pengguna tidak dapat mencari lokasinya saat itu jika aplikasi tidak memiliki fungsi penentuan posisi. Setelah pengguna mendapatkan posisinya, pengguna dapat melakukan navigasi dari lokasi yang telah di dapatkan menuju lokasi yang dituju. Proses pencarian jarak terpendek untuk mendapatkan jalan dari lokasi awal hingga lokasi tujuan memerlukan model simbolik dengan data jarak antar lokasi.

3.7 Use Case Sistem

Use case sistem merupakan diagram kebutuhan yang menggambarkan fungsionalitas sistem dan aktor-aktornya. Berdasarkan pada Gambar 3.4 hanya ada satu aktor terlibat yaitu pengguna. Pengguna berinteraksi dengan *client* untuk menjalankan segala fungsionalitas sistem. *Use case* pada sistem secara umum dijelaskan pada

Gambar 3.4 Diagram *Use Case*

No.	Kode	Nama	Keterangan
1.	UC-01	Melihat Posisi Sekarang	Pengguna dapat melihat lokasinya saat ini
2.	UC-02	Melihat Lokasi yang Tersedia	Pengguna dapat melihat daftar lokasi yang tersedia
3.	UC-03	Melakukan Navigasi ke Lokasi yang Diinginkan	Pengguna dapat melakukan navigasi ke lokasi yang diinginkan dengan memilih lokasi asal dan lokasi tujuan. Tampilan berupa gambar peta yang memiliki panah jalan menuju lokasi yang diinginkan.

Tabel 3.1 Tabel Deskripsi *Use Case*

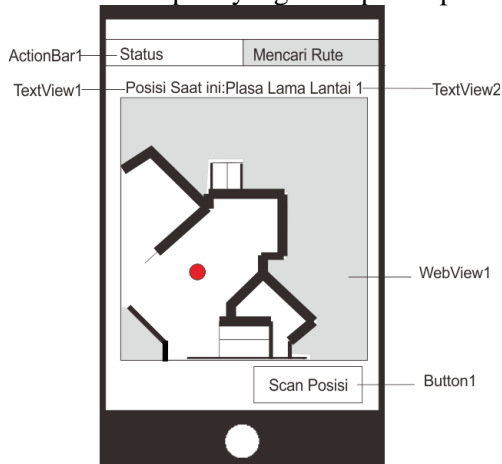
3.8 Perancangan Antarmuka Sistem

Pada Tugas Akhir ini, antarmuka sistem hanya ada pada aplikasi *client*. Rancangan antarmuka sistem meliputi antarmuka melihat posisi saat ini, melihat daftar lokasi yang tersedia dan

navigasi. Antarmuka lokasi saat ini adalah antarmuka yang ditampilkan saat modul aplikasi dijalankan. Antarmuka melihat lokasi yang tersedia adalah antarmuka yang menampilkan semua lokasi. Antarmuka navigasi menampilkan posisi ddari lokasi asal, jalan menuju tujuan dan lokasi tujuan.

3.8.1 Perancangan Antarmuka Melihat Lokasi Saat Ini

Perancangan antarmuka melihat lokasi saat ini terdiri dari beberapa komponen. Ketika aplikasi dibuka, pengguna dapat menekan *button1* untuk melakukan pencarian posisi dengan *scanning* sinyal *Wi-Fi*. Antarmuka ini menampilkan posisi pengguna dengan gambar peta lokasi dan titik posisi pengguna berada. Antarmuka ini terdiri dari *actionbar*, tiga *textview*, satu *listview*, dan satu *button* seperti yang ditampilkan pada Gambar 3.5.



Gambar 3.5 Perancangan Antarmuka Melihat Lokasi Saat Ini

Komponen-komponen yang terdapat pada antarmuka Aduan Terbaru adalah:

- Actionbar1* yang menampilkan nama antarmuka
- Textview1* untuk menampilkan tulisan “Posisi saat ini:”
- Textview2* untuk menampilkan lokasi saat ini

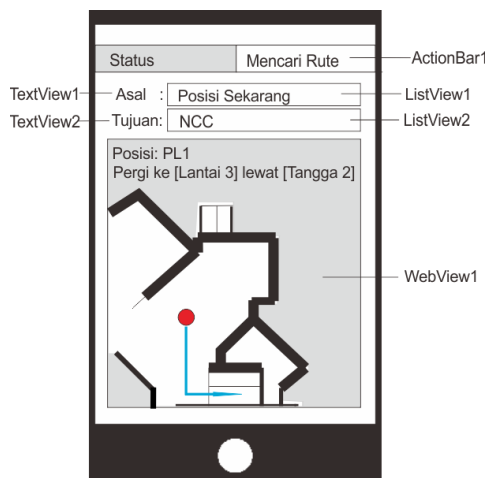
- d. *WebView1* yang berfungsi menampilkan visual lokasi pengguna dan jalan menuju lokasi yang diinginkan jika melakukan navigasi.
- e. *Button1* berfungsi sebagai tombol untuk mengirim hasil *scanning* pengguna dan mengirim permintaan untuk penentuan posisi pengguna

3.8.2 Perancangan Antarmuka Melihat Lokasi yang Tersedia

Antarmuka ini berisi semua lokasi yang tersedia dan menampilkan posisi dari lokasi yang dipilih. Pada antarmuka ini, aplikasi mengakses *database* untuk mendapatkan daftar lokasi yang tersedia.

3.8.3 Perancangan Antarmuka Navigasi

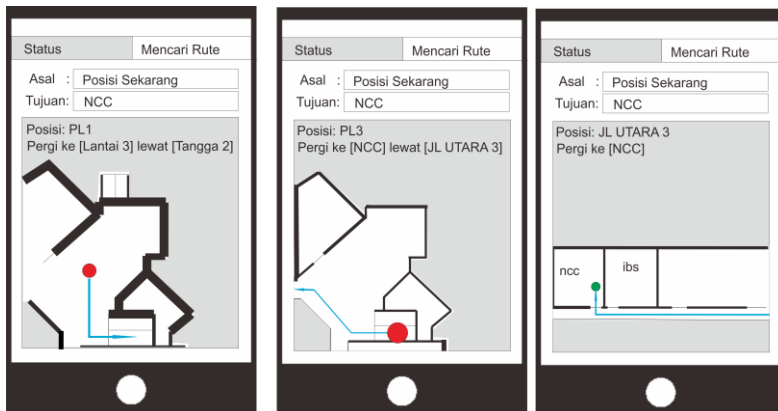
Antarmuka ini berisi semua aduan yang terdapat dalam *database* lokal perangkat bergerak. Pada antarmuka ini juga terdapat komponen yang membantu pengguna dalam mencari suatu aduan seperti yang ditampilkan pada Gambar 3.6.



Gambar 3.6 Perancangan Navigasi

Komponen-komponen yang terdapat pada antarmuka navigasi adalah sebagai berikut:

- Actionbar1* yang menampilkan nama antarmuka
- Textview1* menampilkan tulisan “Asal”.
- Textview2* menampilkan tulisan “Tujuan”.
- ListView1* menampilkan daftar lokasi yang digunakan untuk posisi asal navigasi, perbedaannya dengan lokasi tujuan adalah lokasi asal dapat menampilkan posisi saat ini.
- ListView2* daftar lokasi yang digunakan untuk posisi tujuan navigasi.
- WebView1* menampilkan peta untuk navigasi. Peta untuk navigasi terdiri dari peta posisi asal, peta jalan navigasi dan peta posisi tujuan.



Gambar 3.7 Perancangan Antarmuka Ketika Melakukan Navigasi dari salah satu kasus

Pada Gambar 3.9, Pengguna yang melakukan navigasi akan mendapatkan tampilan peta dengan jalan berupa garis yang menuju ke lokasi dan titik lokasinya saat itu.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Proses implementasi sistem ini dilakukan setelah melewati proses analisis dan perancangan perangkat lunak. Dalam Bab ini akan dibahas mengenai algoritma, *pseudocode* serta implementasi dari perancangan antarmuka yang terdapat dalam perangkat lunak sebagaimana telah dibahas pada Bab III.

4.1 Lingkungan Implementasi

Dalam merancang dan mengimplementasikan perangkat lunak ini, digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1 Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan perangkat lunak ini adalah sebagai berikut:

- Laptop Lenovo G400s,
 - Prosesor Intel® Core™ i3-3120M CPU @ 2.50GHz, dan
 - RAM 8 GB.
- Perangkat *smartphone* Xiaomi Redmi Note 3 Pro
 - CPU: Hexa-core 1.8 GHz Cortex-A72,
 - *Memory* internal: 16 GB,
 - RAM: 2 GB, dan
 - WLAN: Wi-Fi 802.11 b/g/n, Wi-Fi Direct, *hotspot*.

4.1.2 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Microsoft Windows 10 Pro 64-bit sebagai sistem operasi pada laptop,
- Android v7.0 (Nougat) pada *smartphone*,
- Android Developer Tools 23.0.2,

- Menggunakan jasa *Hosting* untuk server dengan paket gratis dari Hostinger.com dengan spesifikasi Intel Xeon, 24 GB shared RAM dan penyimpanan dengan tipe *solid state disk*.
- MySQL 5.6.11 sebagai *database server*,
- Apache 2.4.4 sebagai *web service server*,
- Google Chrome 47.0.2526.111m sebagai *browser* dalam tahap uji coba *server*.
- Draw.io untuk merancang diagram.

4.2 Implementasi Basis Data

Aplikasi ini menggunakan *database* pada *server* berbasis MySQL yang memiliki dua implementasi, yaitu implementasi struktur basis data dan implementasi *query*. Implementasi pada struktur basis data meliputi inisiasi tabel *room* dan *location*. Implementasi *query* meliputi pengolahan data pada *database* seperti *insert* dan *select*.

4.2.1 Implementasi Struktur Basis Data

Implementasi struktur basis data adalah implementasi-implementasi yang menggunakan sintaks dalam pembuatan tabel. Berikut penjelasan masing-masing implementasi setiap tabel.

4.2.1.1 Implementasi Tabel Room

Tabel *room* merupakan tabel yang menyimpan data *Room* di *server*. Implementasi tabel *room* ditunjukkan pada kode 4.1.

```
public static final String CREATE_ROOM =
"CREATE TABLE IF NOT EXISTS " + ROOM + "(" +
    "id_room INTEGER PRIMARY KEY, "
+
    "name TEXT)";
```

Kode 4.1 Implementasi Tabel *Room*

4.2.1.2 Implementasi Tabel Lokasi

Tabel *location* merupakan tabel yang menyimpan semua data latih dari setiap segmen dari lokasi yang tersedia. Implementasi tabel *location* ditunjukkan pada kode 4.2.

```

public static final String CREATE_LOCATION =
"CREATE TABLE IF NOT EXISTS " + LOCATION + "(" +
    "id_location INTEGER PRIMARY KEY
AUTOINCREMENT, " +
    "id_room INTEGER, " +
    "segment INTEGER, " +
    "distances TEXT)";

```

Kode 4.2 Implementasi Tabel *location*

4.2.1.3 Implementasi *locationGraph*

Array locationGraph adalah *array* yang menyimpan hubungan dari sebuah lokasi dengan lokasi lainnya dan termasuk dari pemodelan lokasi simbolik. Penggunaan *array* karena diasumsikan lokasi adalah statis tanpa ada perubahan lebih lanjut. Implementasi *array LocationGraph* ditunjukkan pada kode 4.3.

```

protected $locationsGraph = array(
    'IF101' => ['JALAN UTARA 1'],
    ...
    'PLASA LAMA 3' => ['JALAN UTARA 3', 'RUANG
HIMPUNAN', 'JALAN TIMUR 3', 'TANGGA 2'],
    'PLASA BARU 1' => ['JALAN TIMUR 1', 'JALAN SELATAN
1', 'TANGGA 3'],
    'PLASA BARU 2' => ['JALAN TIMUR 2', 'JALAN SELATAN
2', 'TANGGA 3'],
    'PLASA BARU 3' => ['JALAN TIMUR 3', 'JALAN SELATAN
3', 'TANGGA 3'],
);

```

Kode 4.3 Implementasi *array locationGraph*

4.2.1.4 Implementasi *locationDistance*

Array LocationDistance merupakan *array* yang berisi jarak antara ruangan dengan ruangan yang terhubung dan termasuk dari implementasi pemodelan lokasi simbolik. Urutan pengisian berdasarkan dari *array locationGraph*. Implementasi *array locationDistance* ditunjukkan pada kode 4.4

```
protected $locationsDistance = array(
    'IF101' => [1],
    'IF102' => [1],
    'IF103' => [1],
    ...
    'PLASA LAMA 3' => [2, 1, 2, 2],
    'PLASA BARU 1' => [2, 2, 2],
    'PLASA BARU 2' => [2, 2, 2],
    'PLASA BARU 3' => [2, 2, 2],
);
```

Kode 4.4 Implementasi Array *locationDistance*

4.2.1.5 Implementasi mappedLocation

Array *mappedLocation* adalah *array* yang menyimpan daftar lokasi yang ada dalam sebuah peta dan termasuk implementasi pemodelan lokasi simbolik. Implementasi tabel ini ditunjukkan pada kode 4.5.

```
protected $mappedLocation = array(
    'JU1' => ['MUSHOLLA', 'KANTIN', 'IF101',
             'IF102', 'IF103', 'IF104',
             'IF105A', 'IF105B', 'JALAN UTARA 1'],
    'JU2' => ['RUANG DOSEN LAMA',
             'JALAN UTARA 2'],
    'JU3' => ['KANTIN ATAS', 'RPL', 'NCC',
             'IBS', 'LP', 'JALAN UTARA 3'],
    ...
    'TANGGA 2' => ['TANGGA 2'],
    'TANGGA 3' => ['TANGGA 3'],
);
```

Kode 4.5 Implementasi Array *mappedLocation*

4.2.2 Implementasi Query

Implementasi *query* adalah pengolahan data pada *database* dan sistem. Pengolahan data meliputi *insert*, *update*, dan *select* terhadap *database*. Berikut adalah penjelasan dari implementasi-implementasi tersebut.

4.2.2.1 Implementasi *Insert Room*

Implementasi ini berfungsi untuk memasukkan aduan ke dalam *database*. Implementasi ini ditunjukkan pada kode 4.6.


```

function insertRoom($roomName)
{
    $this->connectDatabase();
    if ($this->db === false) {
        return false;
    }
    $roomName = $this->db-
>real_escape_string($roomName);

    $query = "INSERT INTO room (name) VALUES
('$roomName')";

    $result = $this->db->query($query);
    $this->closeDatabase();
    if ($result > 0) {
        return true;
    }
    return false;
}

```

Kode 4.6 Implementasi *insert Room*

4.2.2.2 Implementasi *Insert Location*

Implementasi ini berfungsi untuk memasukkan kata ke dalam *database*. Implementasi ini ditunjukkan pada kode 4.7.

```

function insertLocation($idRoom, $segment,
$distances)
{
    $this->connectDatabase();
    if ($this->db === false) {
        return false;
    }
    $idRoom = $this->db-
        >real_escape_string($idRoom);
    $segment = $this->db-
        >real_escape_string($segment);
    $distances = $this->db-
        >real_escape_string($distances);

    $query = "INSERT INTO location (id_room,
segment, distances) VALUES ('$idRoom', '$segment',
'$distances')";

```

```

        $result = $this->db->query($query);
        $this->closeDatabase();

        if ($result > 0) {
            return true;
        }
        return false;
    }

```

Kode 4.7 Implementasi *insert Location*

4.2.2.3 Implementasi *Get Room*

Implementasi ini berfungsi untuk mengembalikan nilai *TF* dari suatu *term*. Implementasi ini ditunjukkan pada kode 4.8.

```

function getAllRooms()
{
    $this->connectDatabase();
    if ($this->db === false) {
        return false;
    }

    $query = "SELECT id_room, name FROM room
WHERE is_deleted = 0";

    $result = $this->db->query($query);
    $rooms = [];
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $rooms[] = $row['id_room'] . '_' .
$row['name'];
        }
    }

    $this->closeDatabase();

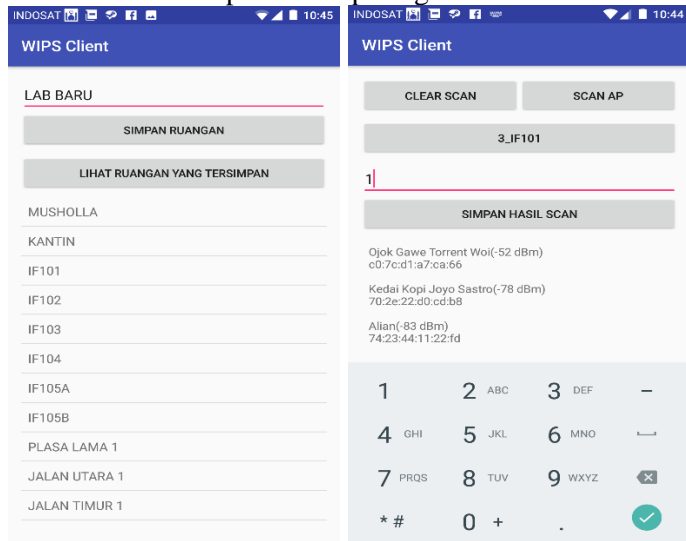
    return $rooms;
}

```

Kode 4.8 Implementasi *get Room*

4.3 Implementasi Pengambilan Data Latih

Sebelum melakukan pencarian posisi maupun navigasi, data latih perlu disiapkan terlebih dahulu. Tampilan dalam Pengambilan data latih dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Antarmuka Pengambilan Data Latih

Sebelum memulai pengambilan data latih, *room* perlu di buat terlebih dahulu. Setelah *room* tersedia, pengambilan data latih dapat dilakukan dengan melakukan *scanning access point* di segmen dari *room* yang diinginkan. Jika pengiriman data latih hasil *scan* gagal, maka aka nada pemberitahuan. Pada antarmuka pengambilan data latih, implementasi *InsertRoom* dan *InsertLocation* digunakan.

4.4 Inspeksi Sampel Data dan Pemilihan Nilai K optimal

Sebelum menentukan posisi, data latih perlu disiapkan agar perhitungan dalam menentukan posisi lebih akurat. Perhitungan dilakukan dengan menerapkan Algoritma *K Nearest Neighbor*

dimana data sampel untuk acuan klasifikasi perlu di inspeksi dahulu dan dibutuhkan nilai K yang optimal agar hasil klasifikasi lebih akurat. Klasifikasi digunakan untuk menentukan posisi pengguna berdasarkan data yang dikirim pengguna dan akan diklasifikasikan untuk menentukan posisi pengguna berdasarkan data sampel yang telah diambil.

Tabel 4.1 adalah hasil perhitungan Euclidean untuk mencari nilai K optimal dengan sistem *voting* dimana untuk pertimbangan dalam pemilihan diberikan pembobotan. Pembobotan yang dilakukan adalah setiap nilai K memiliki nilai awal 100, jika menghasilkan ruangan yang benar dan segmen benar, maka nilai pembobotan tidak dikurangi. Jika menghasilkan ruangan yang benar tetapi segmen salah maka poin akan dikurangi sebanyak 20 poin. Terakhir, jika dari hasil *voting* perhitungan tidak mendapatkan posisi ruangan dan segmen yang akurat, maka poin akan dikurangi sebanyak 100. Jika dalam sebuah K terdapat 2 perhitungan dengan ruangan yang sama, maka akan dilakukan perhitungan rata-rata dari kedua perhitungan dengan ruangan yang sama. Apabila hasil rata-rata masih lebih kecil disbanding hasil rata-rata yang lain, maka nilai terkecil dari ruangan yang sama akan dijadikan acuan.

Ruangan asli	K		
	3	4	5
Ruang14 Segmen4	80	80	80
Ruang15 Segmen1	80	80	80
Ruang14 Segmen4	100	100	100
Ruang7 Segmen2	0	0	0
Ruang13 Segmen4	100	100	100
Ruang4 Segmen1	100	100	100

Ruang4 Segmen3	0	80	0
Ruang1 Segmen1	80	80	80
Ruang1 Segmen4	80	80	80
Ruang6 Segmen3	100	100	100
Ruang9 Segmen2	0	0	0
Ruang2 Segmen1	80	80	80
Ruang5 Segmen2	80	80	80
Total	880	960	880
Rata2	67. 69230769	73. 84615385	67.69 230769

Tabel 4.1 Tabel Perhitungan K dengan voting hasil perhitungan dari Euclidean

Dari tabel 4.1, terlihat bahwa perhitungan dengan $K=4$ dihasilkan data yang lebih akurat dibandingkan $K=3$ dan $K=5$. Penggunaan $K < 3$ tidak digunakan karena tingkat akurasi jauh lebih rendah dan $K > 5$ tidak menghasilkan perhitungan posisi yang optimal.

4.5 Implementasi Menentukan Posisi

Setelah selesai melakukan inspeksi data dan menentukan nilai K yang optimal, aplikasi akan melakukan perhitungan untuk menentukan posisi. Perhitungan kemiripan dilakukan dengan algoritma *K Nearest Neighbor* dengan perhitungan *Euclidean Distance* yang sudah dijelaskan pada subbab 3. Untuk implementasi pendeteksian aduan yang mirip, dapat dilihat pada

```
public static function
getLocation($distancesResult, $k)
{
```

```

        //bagian 1
        asort($distancesResult);
        $kNnResult = array_slice($distancesResult,
0, $k, true);

        //bagian2
        $totalRoom = [];
        $avgDistances = [];
        foreach ($kNnResult as $idLocation =>
$location) {
            $idRoom = $location['id_room'];

            if (isset($totalRoom[$idRoom])) {
                $totalRoom[$idRoom]++;
            } else {
                $totalRoom[$idRoom] = 1;
            }

            if (isset($avgDistances[$idRoom])) {
                $avgDistances[$idRoom] +=
$location['distances'];
                $avgDistances[$idRoom] /=
$totalRoom[$idRoom];
            } else {
                $avgDistances[$idRoom] =
$location['distances'];
            }
        }

        //bagian3
        $maxCountRoom = 0;
        $minAvgDistance = 9999999999999;
        $choosenRoom = 0;
        foreach ($totalRoom as $idRoom =>
$countRoom) {
            if ($totalRoom[$idRoom] > $maxCountRoom)
            {
                $maxCountRoom = $totalRoom[$idRoom];
                $minAvgDistance =
$avgDistances[$idRoom];
                $choosenRoom = $idRoom;
            }
            if ($totalRoom[$idRoom] == $maxCountRoom
&& $minAvgDistance > $avgDistances[$idRoom]) {

```

```

        $maxCountRoom = $totalRoom[$idRoom];
        $minAvgDistance =
$avgDistances[$idRoom];
        $choosenRoom = $idRoom;
    }
}

//bagian4
$finalLocation = [];
foreach ($kNnResult as $idLocation =>
$location) {
    if ($location['id_room'] ==
$choosenRoom) {
        $finalLocation = [
            'room' => $location['id_room'],
            'segment' =>
$location['segment']
        ];
        break;
    }
}

return $finalLocation;
}

```

Kode 4.9 Implementasi dari Menentukan Posisi

Setelah *scanning*, data dikirim kepada *server*. Pertama kali yang dilakukan adalah menghitung jarak *Euclidean* dan memotong jumlah perhitungan sebanyak nilai *K* optimal yang sudah dicari sebelumnya. Selanjutnya, jika ada *Room* yang sama, akan dilakukan perhitungan tambahan untuk menentukan pilihan ruangan yang lebih akurat dengan menghitung rata-rata dari ruangan tersebut.

Proses selanjutnya adalah dari total *room* yang ada, dipilih jumlah *room* terbanyak dengan perhitungan rata-rata paling rendah. Lokasi yang terpilih akan memiliki nilai jarak *Euclidean* yang terendah dan jika terdapat *room* yang sama, nilai rata-rata dari roomnya adalah yang paling rendah.

4.6 Implementasi Navigasi

Implementasi navigasi pada aplikasi ini menggunakan algoritma Dijkstra yang dikembangkan, dimana dalam pencarian jarak, diberi batas sehingga ketika pencarian jarak ke posisi tujuan tidak mencari ke jarak yang tidak optimal. Implementasi ini ditunjukkan pada kode 4.1.

```
public function getNavigation($startLocation,
$endLocation)
{
    $this->resetAll();
    $totalDistance = 0;
    $locationVisited = [$startLocation];
    $this->locationsNode[$startLocation] = 0;
    $this->dijkstra($startLocation,
    $endLocation, $totalDistance,
    $locationVisited);
    $result['total_distance'] = $this-
        >minDistance;
    $result['navigation'] = $this-
        >minLocationVisited;
    return $result;
}
```

Kode 4.10 Implementasi dari Navigasi

4.7 Implementasi Menggambar Posisi dan Navigasi pada Peta

Pada subbab ini dijelaskan implementasi menggambar titik posisi dan jalan untuk navigasi pada gambar peta.

4.7.1 Implementasi Menggambar Posisi

Pada implementasi ini, masukan berupa nama peta yang mencakup sebuah lokasi dan koordinat (x,y) dari posisi yang ditentukan. Keluaran dari implementasi ini adalah gambar berformat PNG dengan titik posisi pada peta lokasi.

```
public static function
getLocationDotImage($roomName, $xPixel, $yPixel)
{
```



```

        $image =
self::loadImageWithSize('img/'.$roomName.'.png',
self::$mapSize[$roomName][0],
self::$mapSize[$roomName][1]);
        $redColor = imagecolorallocate($image, 255,
0, 0);
        imagefilledellipse($image, $xPixel, $yPixel,
15, 15, $redColor);
        return (imagepng($image));
    }

```

Kode 4.11 Implementasi Menggambar Posisi

4.7.2 Implementasi Menggambar Jalan Navigasi

Pada implementasi ini, masukan berupa nama peta yang mencakup sebuah lokasi dan *array* yang berisi nama peta dan lokasi yang terhubung. Keluaran dari implementasi ini adalah gambar berformat PNG dengan jalan berupa garis yang menghubungkan dari sebuah lokasi ke lokasi lain.

```

    public static function
getImageWithNavigation($map, $edges)
    {
        $image =
self::loadImageWithSize('img/'.$map.'.png',
self::$mapSize[$map][0], self::$mapSize[$map][1]);
        $black = imagecolorallocate($image, 0, 0,
0);
        $redColor = imagecolorallocate($image, 255,
0, 0);
        imagesetthickness($image, 3);

        foreach ($edges as $edge) {
            $node1 = $edge[0];
            $node2 = $edge[1];
            imageline($image, $node1[0], $node1[1],
$node2[0], $node2[1], $redColor);
            imagefilledellipse($image, $node2[0],
$node2[1], 10, 10, $black);
        }
        return (imagepng($image));
    }

```

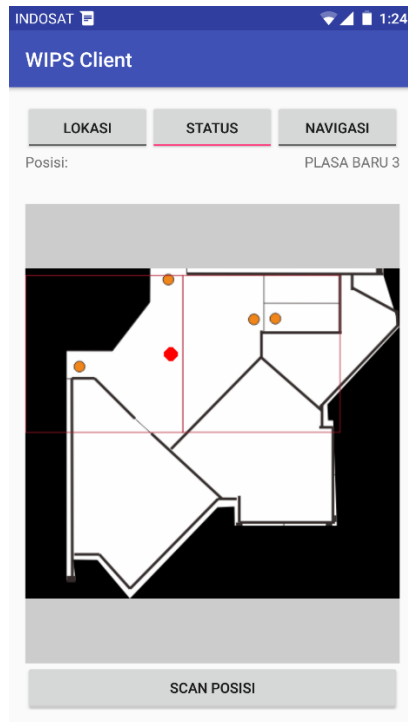
Kode 4.12 Implementasi Menggambar Jalan Navigasi

4.8 Implementasi Antarmuka

Pada subbab ini dijelaskan implementasi tampilan antarmuka yang telah dibahas pada subbab 3.8.

4.8.1 Antarmuka Lokasi Saat Ini

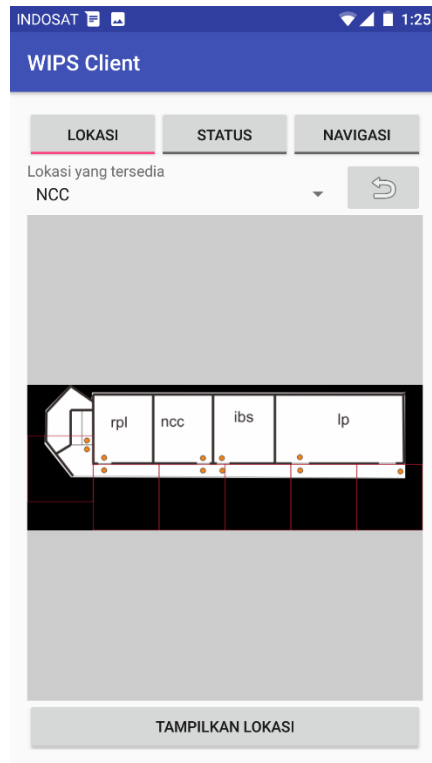
Antarmuka aduan terbaru merupakan antarmuka yang pertama kali ditampilkan ketika *client* dibuka. Pada antarmuka ini dilakukan proses *scanning* untuk mendapatkan kekuatan sinyal wifi yang tersedia di sekitar pengguna. Setelah mendapatkan data kekuatan sinyal *Wi-Fi* dan *mac address* dari *scanning*, kemudian *client* akan mengirimkan data tersebut kepada *server* untuk diproses. Proses yang dilakukan oleh *server* adalah menghitung posisi pengguna dan menggambarkan titik posisi pengguna pada file peta untuk dikirimkan kembali pada *client*. *Client* akan menerima umpan balik dari *server* berupa *file* gambar peta dengan titik posisi dengan format PNG.



Gambar 4.2 Tampilan Antarmuka Lokasi Saat Ini

4.8.2 Antarmuka Daftar Lokasi yang Tersedia

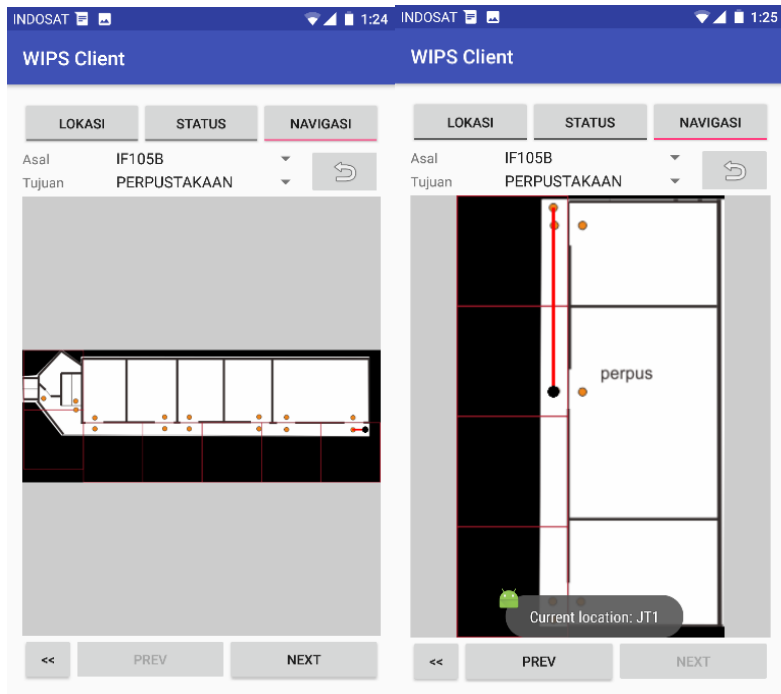
Antarmuka menampilkan semua lokasi yang tersedia di lingkungan jurusan Teknik Informatika ITS. Pengguna dapat memilih lokasi yang diinginkan dan menampilkan gambar peta dari lokasi tersebut. Pada antarmuka ini *client* mengirimkan perintah ke *server* untuk mendapatkan *list* dari lokasi yang tersedia. Ketika pengguna memilih sebuah lokasi dari daftar lokasi, *client* akan mengirimkan perintah untuk menampilkan gambar peta dari lokasi yang dipilih.



Gambar 4.3 Tampilan Antarmuka Lokasi yang Tersedia

4.8.3 Antarmuka Navigasi

Antarmuka Navigasi menampilkan gambar peta yang akan digunakan dalam melakukan navigasi. Pengguna dapat memilih lokasi awal dan lokasi tujuan dari daftar posisi yang diambil dari *server*. Posisi saat ini juga dapat digunakan sebagai lokasi awal untuk melakukan navigasi. Setelah menentukan posisi awal dan posisi akhir, *client* akan mengirimkan parameter tersebut kepada *server*. *Server* kemudian akan mengirimkan gambar peta yang telah diberi penunjuk jalan dari lokasi awal menuju lokasi tujuan. Antarmuka ini menggunakan implementasi fungsi *getNavigation*.



Gambar 4.4 Tampilan Antarmuka Navigasi

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan terhadap sistem yang dibuat. Pembahasan yang dipaparkan meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

5.1 Lingkungan Uji Coba

Proses uji coba terhadap sistem ini dilakukan menggunakan *smartphone* sebagai client, *server* yang mendukung PHP serta MySQL dan lokasi studi kasus gedung bertingkat. Perangkat client yang digunakan adalah *smartphone* Xiaomi Redmi Note 3 Pro dengan spesifikasi seperti berikut:

- Sistem Operasi: Android v7.0 (Nougat),
- CPU: Hexa-core 1.8 GHz Cortex-A72 ,
- *Memory* internal: 16 GB,
- RAM: 2 GB, dan
- WLAN: Wi-Fi 802.11 b/g/n, Wi-Fi Direct, *hotspot*.

Server yang digunakan adalah *server* dari Hostinger,com dengan paket domain dan hosting gratis. Spesifikasi *server* yang digunakan adalah sebagai berikut:

- CPU: Intel Xeon
- RAM: 24 GB Shared
- Jenis *Memory*: Solid State Disk

Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember akan digunakan sebagai lokasi dari kasus gedung bertingkat. Uji coba dilakukan dengan cara menjalankan aplikasi dan menjalankan fungsional dari aplikasi.

5.2 Data Uji Coba

Data uji coba yang digunakan adalah list kekuatan sinyal Wi-Fi dari proses *scanning* yang dilakukan oleh pengguna.

5.3 Skenario & Hasil Uji Coba

Pada subbab ini dijelaskan skenario uji coba yang dilakukan dan hasil yang didapatkan. Uji coba terdiri dibagi menjadi empat yaitu uji coba fungsionalitas, uji coba pengambilan dan pemrosesan data, uji coba tingkat akurasi menentukan posisi, dan uji coba melakukan navigasi.

5.3.1 Pengujian Fungsionalitas

Uji coba fungsionalitas merupakan sebuah pengujian yang dilakukan terhadap jalannya fungsi-fungsi utama pada sistem yang telah dibuat. Uji coba fungsionalitas dilakukan terhadap semua *use case* yang telah dijelaskan pada subbab 3.5.

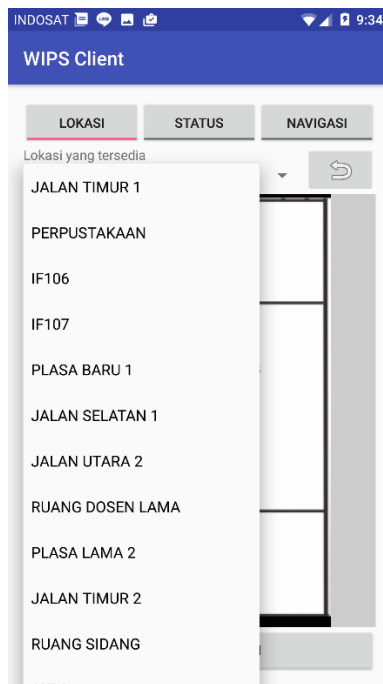
5.3.1.1 Uji Coba Menampilkan Lokasi yang Tersedia

Uji coba ini dilakukan dengan mencoba menampilkan lokasi yang tersedia. Uji coba dilakukan dengan dua skenario seperti yang tertera pada Tabel 5.1.

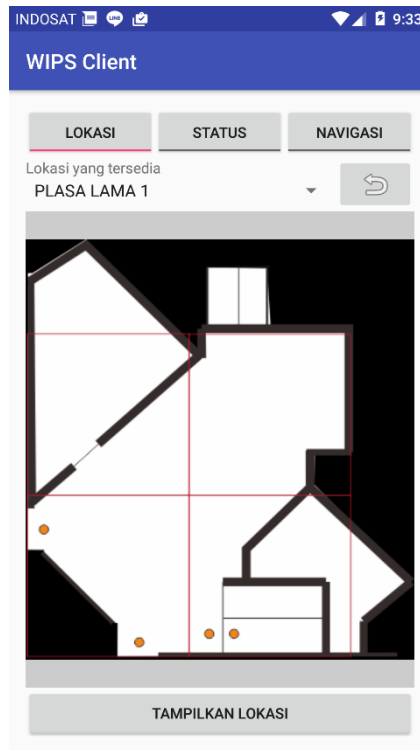
Tabel 5.1 Prosedur Uji Coba Lokasi yang Tersedia

ID	UJ-01
Referensi Use Case	UC-01
Nama	Uji Coba Menampilkan Lokasi yang Tersedia
Tujuan Uji Coba	Menguji fitur untuk menampilkan lokasi yang tersedia.
Kondisi Awal	-.
Skenario 1	Pengguna ingin melihat lokasi yang ada di lingkungan Teknik Informatika ITS
Masukan	-

Keluaran	Aplikasi menampilkan list dari 45 lokasi yang tersedia.
Hasil Uji Coba	Berhasil
Skenario 2	Pengguna ingin melihat gambar peta dari lokasi yang diinginkan
Masukan	-
Keluaran	Aplikasi menampilkan gambar peta dari lokasi yang diinginkan
Hasil Uji Coba	Berhasil



Gambar 5.1 Hasil Uji Coba Menampilkan Lokasi yang Tersedia

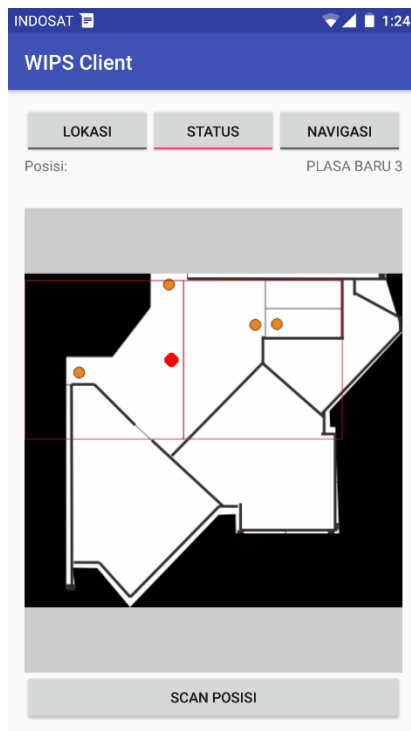


Gambar 5.2 Tampilan Uji Coba Melihat Gambar Peta dari Lokasi yang Diinginkan

Sesuai dengan prosedur pada Tabel 5.1, ketika pengguna menjalankan aplikasi, pengguna akan memasuki antarmuka Lokasi. Pada antarmuka ini, aplikasi dapat menampilkan 45 lokasi yang tersedia pada daftar seperti pada Gambar 5.1. Pada antarmuka ini juga terdapat tombol “TAMPILKAN LOKASI”. Tombol ini berfungsi untuk melihat peta dari daftar lokasi yang diinginkan seperti pada Gambar 5.2.

5.3.1.2 Uji Coba Menampilkan Lokasi Saat Ini

Uji coba ini dilakukan dengan menekan tombol “SCAN POSISI”. Pengguna cukup berada di sebuah lokasi yang ada di Lingkungan Jurusan Teknik Informatika ITS, kemudian pengguna menekan tombol “SCAN POSISI”. Setelah menekan tombol, *client* akan melakukan *scanning* data kekuatan sinyal *wi-fi* dari sekitar beserta *mac address* nya dan mengirim ke server untuk diproses. Server akan mengirim gambar berupa peta dan titik lokasi dari pengguna seperti pada Gambar 5.3.



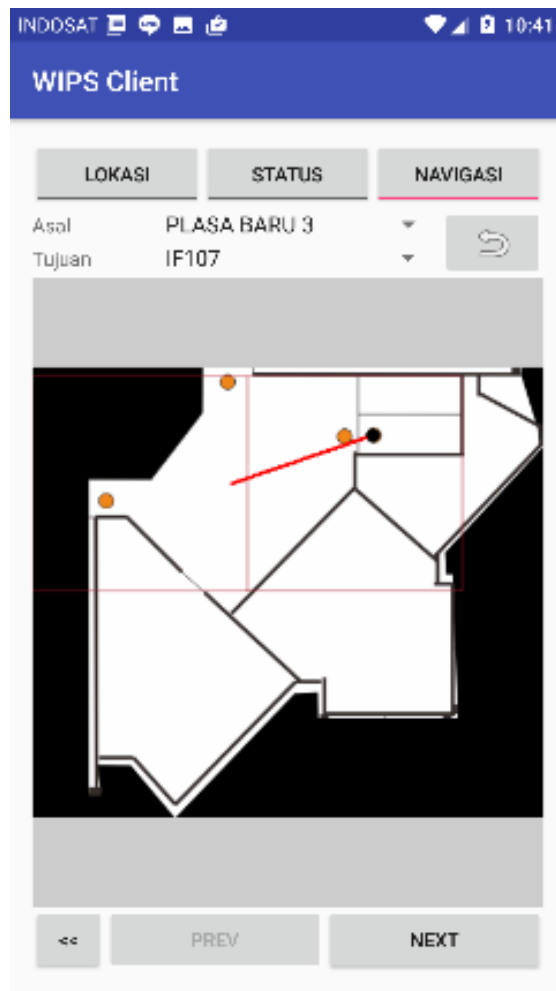
Gambar 5.3 Hasil Uji Coba Menampilkan Lokasi Saat Ini

Tabel 5.2 Prosedur Uji Coba Menampilkan Lokasi Saat Ini

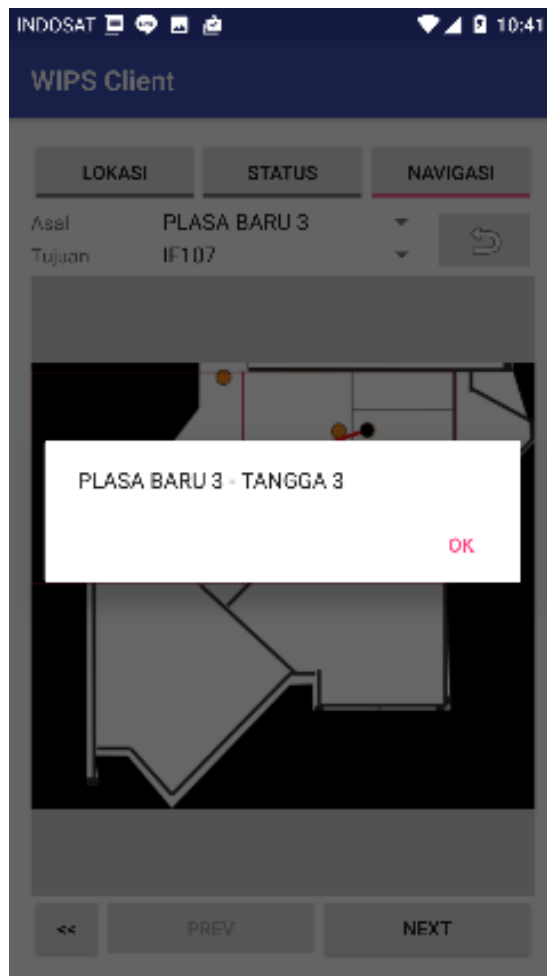
ID	UJ-02
Referensi Use Case	UC-02
Nama	Uji Coba Menampilkan Lokasi Saat Ini
Tujuan Uji Coba	Menguji fitur menampilkan semua aduan
Kondisi Awal	Pengguna berada pada lokasi yang ada di lingkungan Jurusan Teknik Informatika ITS
Skenario	Pengguna menekan tombol “SCAN POSISI”
Masukan	Kekuatan sinyal wifi yang tersedia di lokasi tersebut
Keluaran	Peta lokasi saat ini
Hasil Uji Coba	Berhasil

5.3.1.3 Uji Coba Melakukan Navigasi

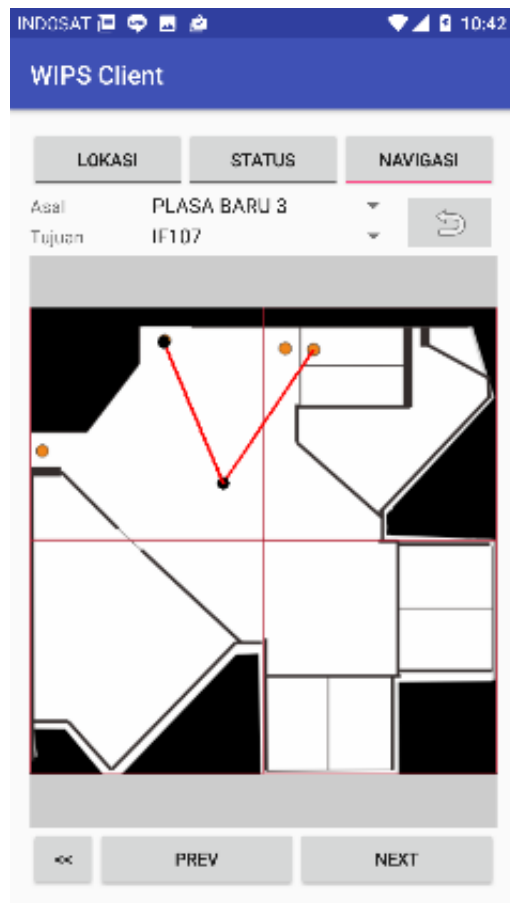
Uji coba ini dilakukan dengan memilih lokasi awal dan lokasi tujuan dari daftar lokasi yang tersedia. Prosedur dalam uji coba ini dijelaskan pada Tabel 5.3. Uji coba dilakukan dengan masuk ke antarmuka navigasi di ujung kanan atas tampilan. Informasi yang ditampilkan pada antarmuka navigasi adalah peta lokasi dengan jalan untuk navigasi menuju ke lokasi tujuan. Pengguna juga dapat menggunakan posisi saat ini untuk posisi awal dari navigasi. Hasil dari uji coba dengan posisi awal Plasa Baru Lantai 3 ke posisi akhir IF107 dapat dilihat pada Gambar 5.8.



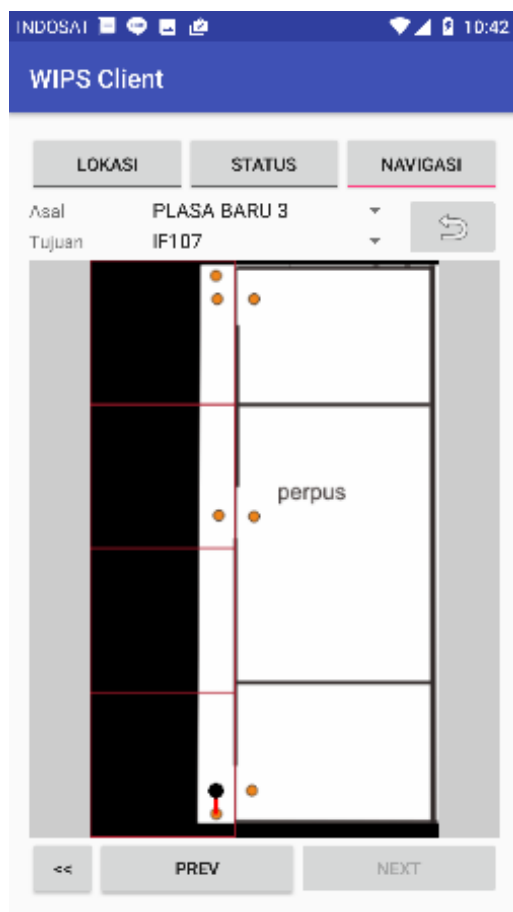
Gambar 5.4 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107



Gambar 5.5 Tampilan Detil Peta Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107



Gambar 5.6 Tampilan Jalan Tangga 3 ke Jalan Timur Lantai 1 Melalui Plasa Baru 1 pada Antarmuka Navigasi dari Plasa Baru Lantai 3 Menuju IF107



Gambar 5.7 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru 3 Menuju IF107



Gambar 5.8 Tampilan Jalan Plasa Baru Lantai 3 ke Tangga 3 pada Antarmuka Navigasi dari Plasa Baru 3 Menuju IF107

Tabel 5.3 Prosedur Uji Coba Melakukan Navigasi

ID	UJ-03
Referensi Use Case	UC-03
Nama	Uji Coba Melakukan Navigasi

Tujuan Uji Coba	Menguji fitur melakukan navigasi
Kondisi Awal	Pengguna berada dalam antarmuka navigasi
Skenario	Pengguna ingin melakukan navigasi.
Masukan	Posisi awal dan posisi akhir yang dipilih dari list yang tersedia
Keluaran	Gambar peta lokasi dengan garis yang menunjukkan jalan yang optimal dari posisi awal menuju posisi tujuan
Hasil Uji Coba	Berhasil

5.3.2 Uji Coba Akurasi Deteksi Posisi

Uji coba ini dilakukan dengan melakukan *scanning* di tempat yang telah ditentukan. Hasil dari deteksi posisi akan dicatat dan kemudian dikumpulkan untuk melihat persentase dari akurasi deteksi posisi. Cakupan deteksi lokasi berada di tingkat ruangan atau *room*.

Uji Coba ke-	Posisi Scanning	Posisi Hasil Scanning
1	LP	JALAN UTARA 3
2	LP	LP
3	LP	LP
4	LP	LP
5	PLASA LAMA 3	PLASA LAMA 3
6	PLASA LAMA 3	PLASA LAMA 3
7	PLASA LAMA 3	PLASA LAMA 3
8	PLASA LAMA 3	PLASA LAMA 3
9	PLASA BARU 3	PLASA BARU 3
10	PLASA BARU 3	JALAN TIMUR 3

11	PLASA BARU 3	PLASA BARU 3
12	PLASA BARU 3	PLASA BARU 3
13	TANGGA 2	TANGGA 2
14	TANGGA 2	TANGGA2
15	LP2	LP2
16	LP2	PLASA LAMA 3
17	LP2	LP2
18	LP2	LP2
19	JALAN TIMUR 1	JALAN TIMUR 1
20	JALAN TIMUR 1	JALAN TIMUR 1
21	JALAN TIMUR 1	JALAN TIMUR 3
22	PLASA LAMA 2	PLASA LAMA 2
23	PLASA LAMA 2	PLASA LAMA 2
24	PLASA LAMA 2	PLASA LAMA 2
25	RUANG DOSEN BARU	JALAN UTARA 2
26	RUANG DOSEN BARU	RUANG DOSEN BARU
27	RUANG DOSEN BARU	RUANG DOSEN BARU
28	PLASA BARU 1	PLASA BARU 1
29	PLASA BARU 1	PLASA BARU 1
30	PLASA BARU 1	PLASA BARU 1

Tabel 5.4 Uji Coba Akurasi Deteksi Posisi

Dari uji coba tabel 5.4, didapatkan bahwa 26 dari 30 percobaan berhasil mendapatkan ruangan yang tepat. Lokasi jalan selatan baik lantai 1 hingga 3 tidak berhasil mendapatkan hasil *scan* yang akurat. Secara persentase, tingkat akurasi dari menampilkan lokasi saat ini adalah 86.66%.

5.3.3 Uji Coba Performa dari Aplikasi

Uji coba ini dilakukan untuk menghitung waktu yang dibutuhkan untuk memproses data pada *server*. Dalam uji coba ini, akan dilakukan performa mencari posisi pengguna dan navigasi.

5.3.3.1 Uji Coba Performa Mencari Posisi saat Ini

Uji coba ini dilakukan dengan menghitung waktu pengolahan data oleh *server* ketika mencari posisi saat ini. Kecepatan dihitung ketika aplikasi menerima data berupa kekuatan sinyal *Wi-Fi* sampai menghasilkan output gambar dengan titik posisi pengguna. Satuan waktu yang digunakan untuk menghitung kecepatan dalam mengolah data adalah detik (s). Uji coba akan dilakukan sebanyak 10 kali dengan data kekuatan sinyal *Wi-Fi* yang berbeda-beda.

Percobaan ke-	Waktu Sebelum (s)	Waktu Sesudah (s)	Selisih Waktu (s)
1	0.714607	0.749195	0.034599
2	0.944472	0.97113	0.026668
3	0.871145	0.903127	0.031996
4	0.132743	0.159457	0.026729
5	0.235939	0.270632	0.034708
6	0.191201	0.216205	0.025019
7	0.400406	0.425333	0.024941
8	0.758401	0.782684	0.024299
9	0.059673	0.083372	0.023713
10	0.24448800	0.27085700	0.026383

Tabel 5.5 Uji Coba Performa Mencari Posisi saat Ini

5.3.3.2 Uji Coba Performa Melakukan Navigasi

Uji coba ini dilakukan dengan menghitung waktu pengolahan data oleh *server* ketika melakukan navigasi. Kecepatan dihitung ketika aplikasi menerima parameter lokasi awal dan lokasi tujuan sampai aplikasi menampilkan gambar dengan jalan navigasi. Satuan waktu yang digunakan untuk menghitung kecepatan dalam mengolah data adalah detik (s). Uji coba akan dilakukan sebanyak

10 kali dengan parameter lokasi awal dan lokasi tujuan yang berbeda-beda.

Percobaan ke-	Waktu Sebelum (s)	Waktu Sesudah (s)	Selisih Waktu (s)
1	0.04628000	0.04738000	0.001114
2	0.40636900	0.40759200	0.001236
3	0.27206000	0.27306400	0.001014
4	0.46788600	0.46827800	0.0004009
5	0.37130500	0.37202700	0.000728
6	0.7791200	0.78013500	0.001023
7	0.10947100	0.11106400	0.001608
8	0.57685900	0.57769800	0.00084399
9	0.92806800	0.92926200	0.001207
10	0.47305000	0.48856800	0.015538

Tabel 5.6 Uji Coba Performa Melakukan Navigasi

5.4 Evaluasi

Dari uji coba yang telah dilakukan, diketahui bahwa sistem telah dapat bekerja dengan baik dan benar dalam menjalankan fungsionalitasnya. Evaluasi pengujian pada masing-masing pengujian dijelaskan sebagai berikut.

1. Pengujian operasional aplikasi berjalan dengan baik dan sesuai dengan yang diharapkan. Tidak ada *use-case* yang gagal dijalankan dengan beberapa skenario yang dicoba.
2. Pengujian dalam pengambilan dan melihat data latih berjalan dengan baik dan dapat menampilkan informasi dari hasil *scanning* yaitu nama *access point*, kekuatan sinyal dan *mac address*.
3. Pengujian akurasi deteksi posisi terlihat bahwa aplikasi dapat mendapatkan posisi akurat sebanyak 26 dari 30 percobaan atau 86.66% tingkat keberhasilan.
4. Pengujian performa dari aplikasi menghasilkan waktu rata-rata 23ms (milisecond). Untuk pencarian navigasi, waktu proses rata-rata adalah 1ms (milisecond).

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Dalam bab terakhir ini dijelaskan kesimpulan yang didapat dari pengerjaan Tugas Akhir beserta saran-saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut.

6.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, terdapat beberapa kesimpulan yang bisa diambil, yaitu:

1. Pemodelan lokasi untuk gedung bertingkat dengan model simbolik dapat digunakan. Fitur dari pemodelan seperti lokasi terhubung, lokasi yang mengandung dan jarak antar lokasi dapat dimodelkan.
2. Navigasi dapat menampilkan jalan yang optimal untuk menuju ke lokasi tujuan.
3. Tingkat akurasi untuk deteksi posisi sudah cukup baik. Tingkat akurasi 86.66% dapat diandalkan untuk penggunaan secara umum dengan ruangan yang memiliki *access point* yang dipasang dengan rapih.
4. Pada uji performa, *server* dari aplikasi dapat melakukan proses dengan cepat yaitu sekitar 28ms (milisecond) untuk pencarian posisi dan 1ms (milisecond) untuk pencarian navigasi. Meskipun ujicoba performa dapat dilakukan dengan cepat, kondisi jaringan dapat mempengaruhi *client* dalam mengirim data dan mendapatkan respon dari *server*.

6.2 Saran

Adapun saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut adalah sebagai berikut:

1. Melakukan Clustering pada data sinyal Wi-Fi untuk mendapatkan hasil posisi yang lebih akurat [3]. Selain itu,

perlu memperbanyak data latih dan kualitasnya untuk setiap segmen.

2. Memasang *Access Point* dengan lebih teratur dan rapih sehingga data sinyal wifi untuk setiap posisi lebih baik. Data sinyal *Wi-Fi* yang teratur akan menghasilkan perhitungan posisi yang lebih akurat.
3. Melakukan pengukuran jarak antar lokasi pada model secara lebih detil, sehingga keakuratan navigasi akan menjadi lebih baik.
4. Tampilan antarmuka dibuat lebih mudah untuk pengguna dan peta ditampilkan dalam bentuk yang lebih baik.
5. Menggunakan pendekatan lainnya untuk melakukan deteksi posisi pengguna seperti *Bayesian Statistical Model* untuk pendekatan secara statistik.
6. Pada kasus gedung bertingkat, *server* sebaiknya disimpan pada lokal dibandingkan *hosting* di internet. Meskipun terdapat sinyal *Wi-Fi* atau koneksi data *smartphone* untuk mengirim dan menerima respon dari *server*, ketika berada di dalam gedung, tetapi tidak ada jaminan untuk akses koneksi yang stabil atau dapat diandalkan.

DAFTAR PUSTAKA

- [1] I. Satoh, "A location model for smart environments," *Design and Use of Smart Environments*, vol. 3, no. 2, p. 158–179, 2007.
- [2] A. K. Dey, A. Schmidt dan J. F. McCarthy, "UbiComp 2003: Ubiquitous Computing," dalam *UbiComp 2003, the 5th Annual Conference on Ubiquitous Computing*, Seattle, 2003.
- [3] R. I. P. Putra, Rancang Bangun Sistem Pendeteksi Posisi dalam Ruangan Menggunakan Kekuatan Sinyal Wi-Fi dengan Penerapan Algoritma Cluster Filtered KNN, Surabaya, 2013.
- [4] S. Loke, "Context Aware Pervasive Sytem: Architectures of a New Breed of Applications," New York, Auerbach Pub, 2006, p. 7.
- [5] "Wi-Fi - IEEE Standards," Tutorialspoint, [Online]. Available: http://www.tutorialspoint.com/wi-fi/wifi_ieee_standards.htm. [Diakses 3 July 2015].
- [6] S. Sayad, "An Introduction to Data Mining: K Nearest Neighbors - Classification," [Online]. Available: http://www.saedsayad.com/k_nearest_neighbors.htm. [Diakses 20 5 2017].
- [7] Techterms, January 2010. [Online]. Available: <http://techterms.com/definition/android>.
- [8] "Android Documentation: WifiManager," [Online]. Available:

<https://developer.android.com/reference/android/net/wifi/WifiManager.html>. [Diakses 20 5 2017].

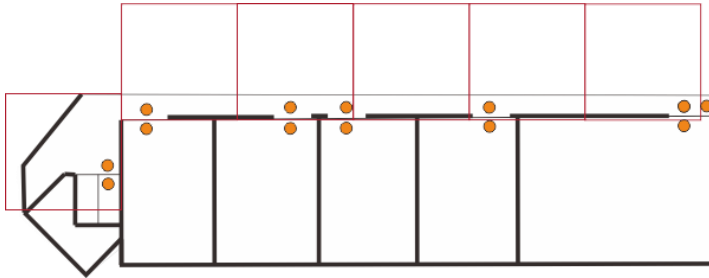
- [9] “PHP: Hypertext Preprocessor,” [Online]. Available: <http://php.net/>. [Diakses 20 May 2017].
- 10] W. Hastomo, “Pengertian dan Kelebihan Database MySQL,” [Online]. Available: <http://hastomo.net/php/pengertian-dan-kelebihan-database-mysql/>. [Diakses 20 May 2017].

LAMPIRAN A

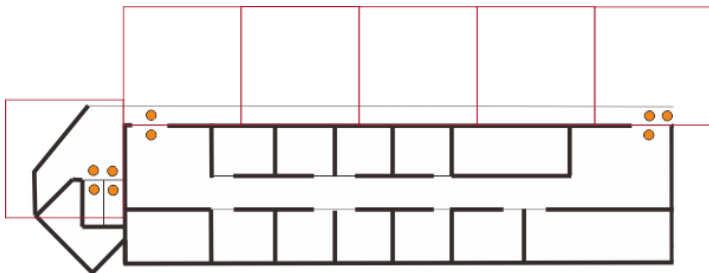
DAFTAR LOKASI YANG DIGUNAKAN

Lampiran ini berisi peta dari lokasi-lokasi yang digunakan. Setiap peta memiliki pembagian segmen yang digunakan untuk menentukan posisi pengguna berdasarkan sinyal Wi-Fi yang sesuai dengan segmen

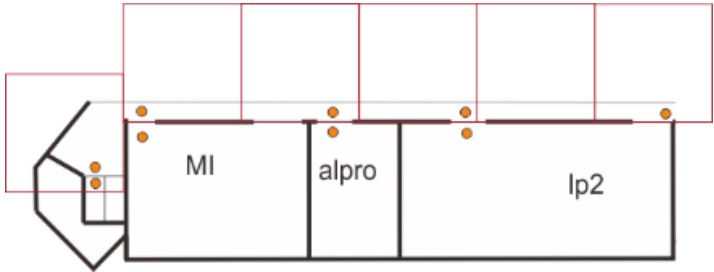
A. Peta Lokasi yang Digunakan



Jalan Selatan Lantai 1



Jalan Selatan Lantai 2



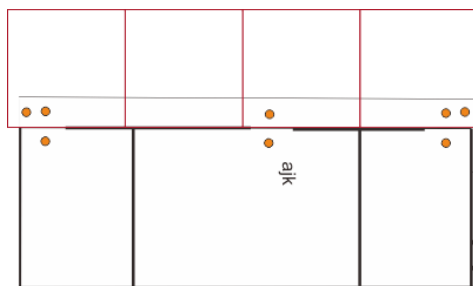
Jalan Selatan Lantai 3



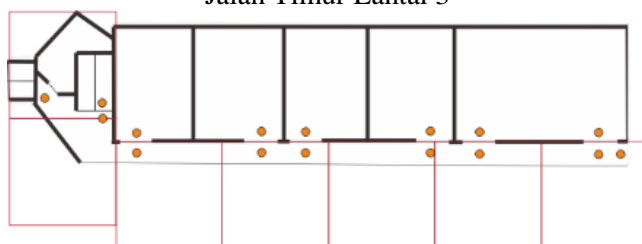
Jalan Timur Lantai 1



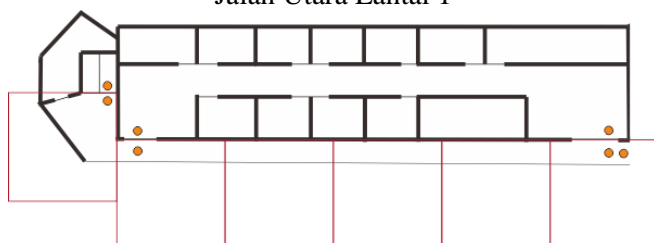
Jalan Timur Lantai 2



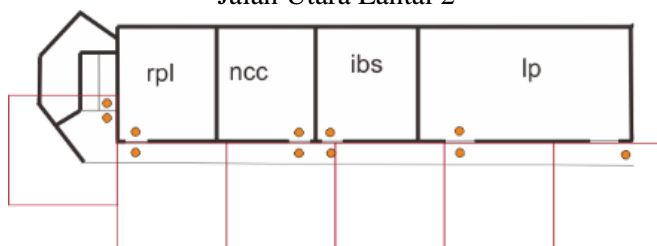
Jalan Timur Lantai 3



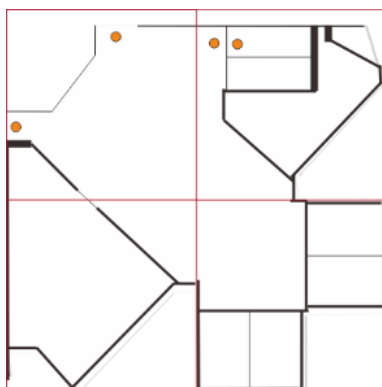
Jalan Utara Lantai 1



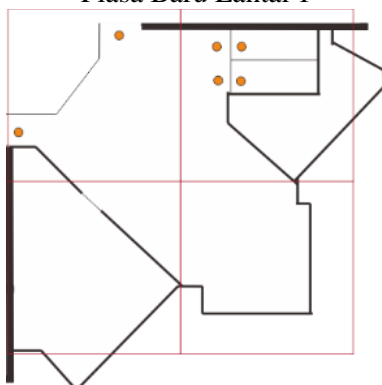
Jalan Utara Lantai 2



Jalan Utara Lantai 3



Plasa Baru Lantai 1



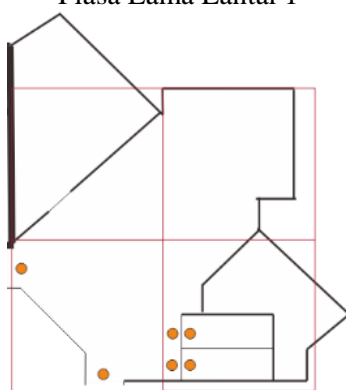
Plasa Baru Lantai 2



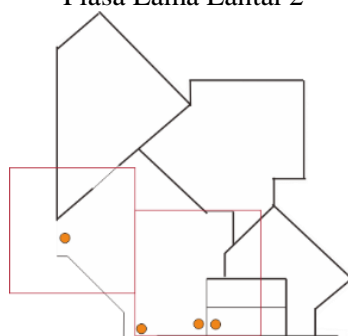
Plasa Baru Lantai 3



Plasa Lama Lantai 1



Plasa Lama Lantai 2



Plasa Lama Lantai 3

B. Daftar Lokasi yang Digunakan

1. MUSHOLLA
2. KANTIN
3. IF101
4. IF102
5. IF103
6. IF104
7. IF105A
8. IF105B
9. PLASA LAMA 1
10. JALAN UTARA 1
11. JALAN TIMUR 1
12. PERPUSTAKAAN
13. IF106
14. IF107
15. PLASA BARU 1
16. JALAN SELATAN 1
17. TANGGA 1
18. TANGGA 2
19. TANGGA 3
20. JALAN UTARA 2
21. RUANG DOSEN LAMA
22. PLASA LAMA 2
23. JALAN TIMUR 2
24. RUANG SIDANG
25. AULA
26. PLASA BARU 2
27. JALAN SELATAN 2
28. RUANG DOSEN BARU
29. KANTIN ATAS
30. JALAN UTARA 3
31. RPL
32. NCC
33. IBS
34. LP

- 35. PLASA LAMA 3
- 36. RUANG HIMPUNAN
- 37. JALAN TIMUR 3
- 38. GAMEDEV
- 39. AJK
- 40. IGS
- 41. PLASA BARU 3
- 42. JALAN SELATAN 3
- 43. LP2
- 44. ALPRO
- 45. MI

LAMPIRAN B
PERHITUNGAN DALAM MENENTUKAN K
YANG OPTIMAL

Pada subbab ini dilampirkan perhitungan dalam menentukan K yang optimal.

- 1. Perhitungan Nilai K optimal
Berikut adalah perhitungan secara detil dari pencarian nilai K yang optimal.

Ruangan asli	Hasil Perhitungan Jarak Euclidean	K			Voting Ruangan
		3	4	5	
Room 14 Segmen 4	[Room 14 Segment 3] 40.804411526206	80			K=3, Room 14
	[Room 14 Segment 2] 46.808118953874	R14 S3			K=4, Room 14
	[Room 14 Segment 4] 47.042533945356				K=5, Room 14
	[Room 13 Segment 4] 56.58621740318		80		
			R14 S3		
	[Room 13 Segment 2] 57.558665724633			80	
			R14 S3		
Room 15 Segmen 1	[Room 15 Segment 2] 51.264022471905	80			K=3, Room 15

	[Room 15 Segment 1] 51.526692111953	R15 S2			K=4, Room 15
	[Room 9 Segment 3] 66.775744099186				K=5, Room 15
	[Room 12 Segment 6] 71.119617546778		80		
			R15 S2		
	[Room 9 Segment 4] 74.067536748565			80	
				R15 S2	
Room 14 Segmen 4	[Room 14 Segment 4] 45.343136195019	100			K=3, Room 14
	[Room 1 Segment 2] 51.429563482495	R14 S4			K=4, Room 14
	[Room 14 Segment 3] 54.221766846904				K=5, Room 14
	[Room 1 Segment 1] 56.982453439634		100		
			R14 S4		
	[Room 14 Segment 2] 58.514955353311			100	
				R14 S4	
Room 7 segmen 2	[Room 1 Segment 4] 77.94870107962	0			K=3, Room 1

	[Room 1 Segment 3] 82.109682742049	R1 S4			K=4, Room 1
	[Room 10 Segment 6] 82.395388220458				K=5, Room 1
	[Room 7 Segment 2] 83.444592395194		0		
			R1 S4		
	[Room 11 Segment 6] 83.725742755738			0	
				R1 S4	
Room 13 segmen 4	[Room 13 Segment 4] 64.039050586341	100			K=3, Room 13
	[Room 14 Segment 4] 64.124878167526	R13 S4			K=4, Room 13
	[Room 13 Segment 2] 66.498120274185				K=5, Room 13
	[Room 13 Segment 3] 68.898476035396		100		
			R13 S4		
	[Room 1 Segment 2] 69.043464571239			100	
				R13 S4	
Room 4 segmen 1	[Room 4 Segment 1] 57.740800133008	100			K=3, Room 4

	[Room 4 Segment 2] 70.334913094423	R4 S1			K=4, Room 4
	[Room 11 Segment 1] 79.655508284111				K=5, Room 4
	[Room 13 Segment 2] 80.993826925266		100		
			R4 S1		
	[Room 4 Segment 3] 81.129526067887			100	
				R4 S1	
Room 4 segmen 3	[Room 7 Segment 2] 75.683551713698	0			K=3, Room 7
	[Room 10 Segment 4] 76.857010090167	R7 S2			K=4, Room 14
	[Room 4 Segment 1] 77.871689335727				K=5, Room 14
	[Room 4 Segment 2] 78.860636568569		80		
			R4 S1		
	[Room 10 Segment 3] 79.730797061111			0	
				R10 S4	
Total Nilai		460	540	460	Iterasi: 7
Rata-Rata		65.71	77.14	65.71	

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Faris Muhammad Asmawi, lahir di Bandung, pada tanggal 21 April 1993. Penulis adalah anak pertama dari tiga besaudara. Pendidikan formal yang telah ditempuh penulis adalah TK Al Fajar (1998-1999), MI Asih Putera (1999-2004), SMP Darul Hikam

(2005-2008), SMAN 3 Bandung (2008-2011). Setelah lulus dari SMAN 3 Bandung, penulis diterima di jurusan Teknik Informatika ITS angkatan 2011 dengan NRP 5111100157. Selama menempuh kuliah, penulis juga aktif berperan di organisasi dan komunitas kampus diantaranya sebagai Staf Pengembangan Profesi Himpunan Mahasiswa Teknik Komputer Informatika ITS (HMTK-ITS) C-1B, Staf Student Resource Development BEM Fakultas Teknologi Infomasi ITS, dan Administrator dari Net Centric Computing Laboratory. Penulis dapat dihubungi melalui alamat email faris.asmawi@gmail.com.